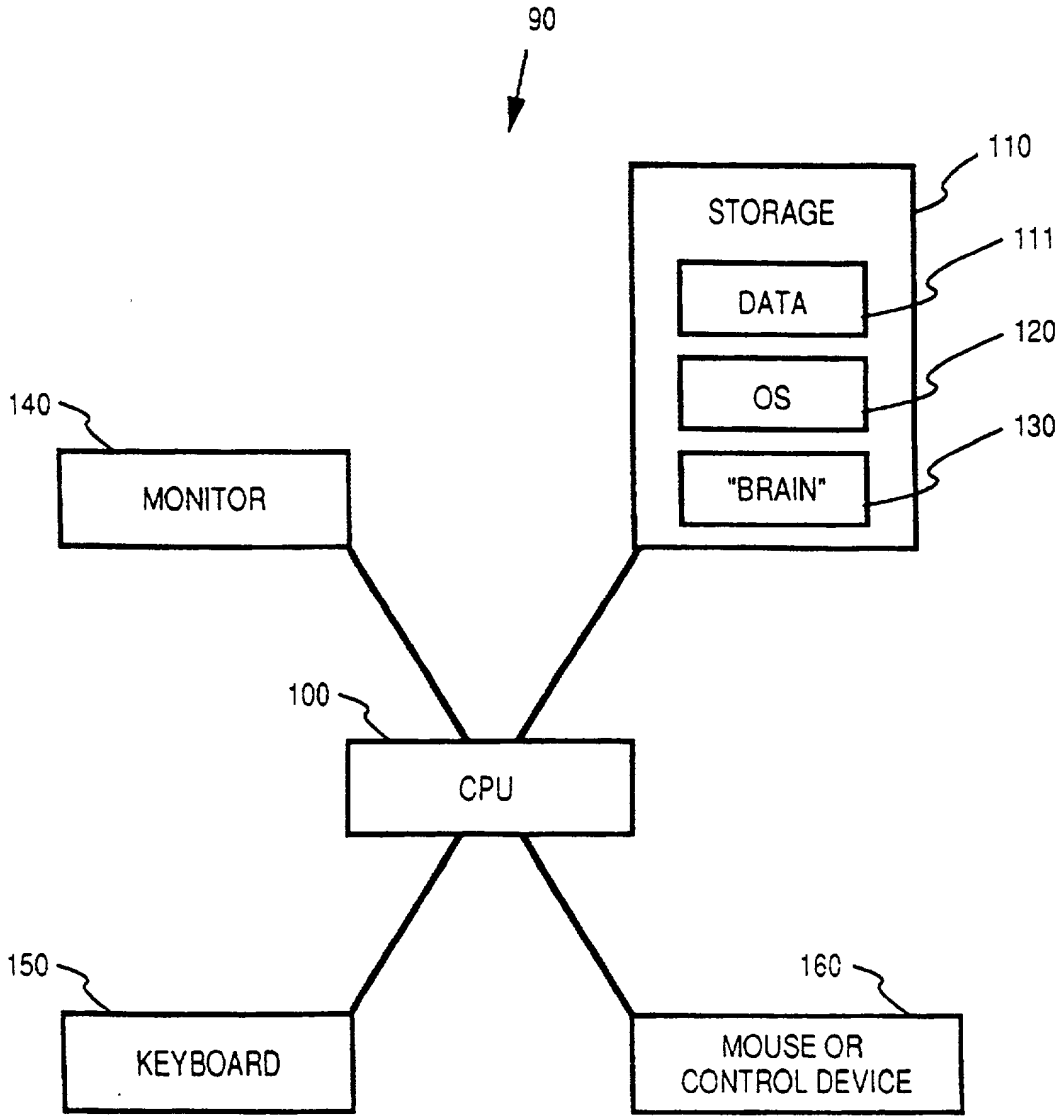

90
110
STORAGE
111
DATA
120
OS
130
"BRAIN"
140
MONITOR
100
CPU
150
KEYBOARD
160
MOUSE OR
CONTROL DEVICE

Fig.1

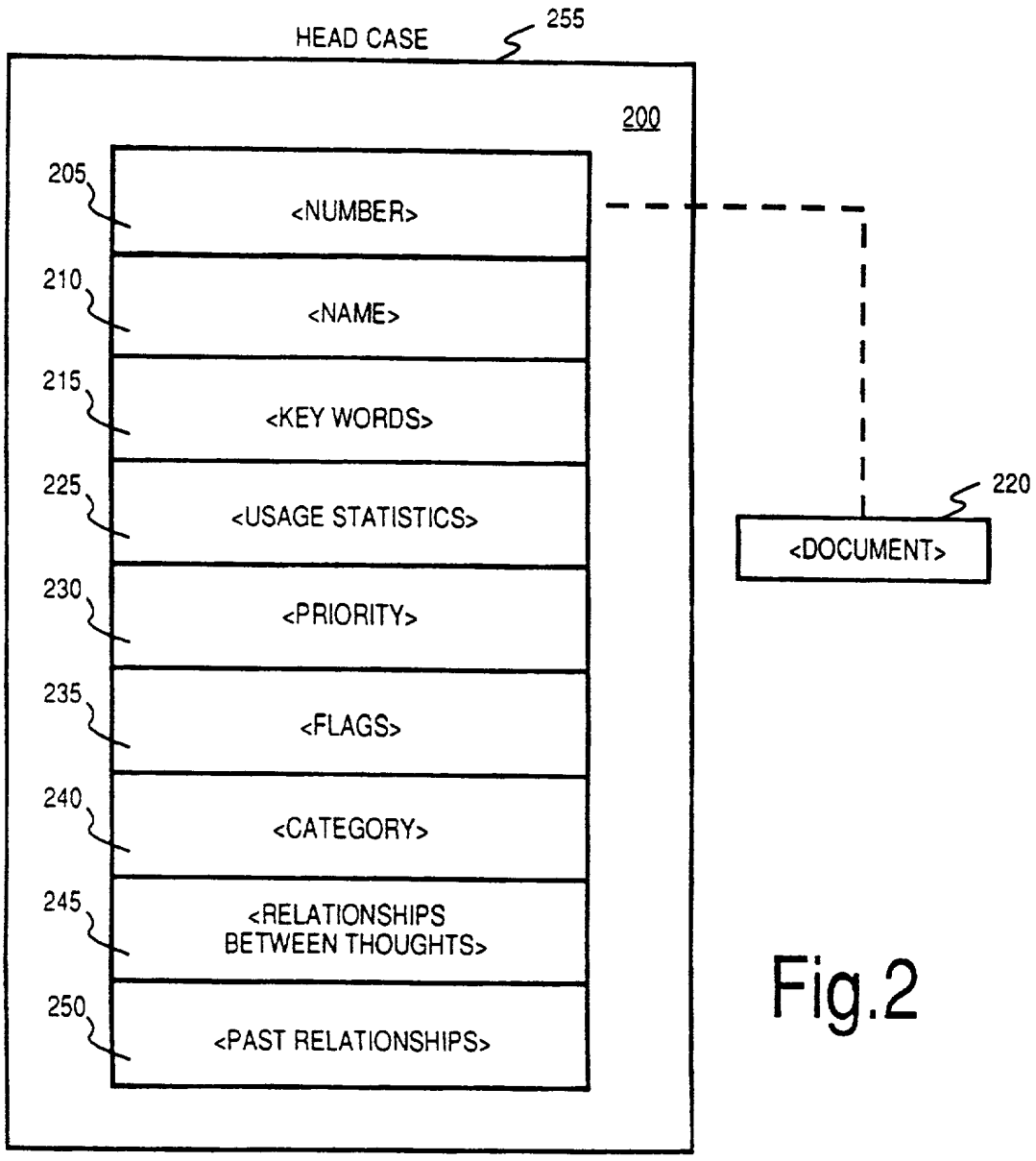

HEAD CASE
255
200
205
<NUMBER>
210
<NAME>
215
<KEY WORDS>
225
<USAGE STATISTICS>
230
<PRIORITY>
235
<FLAGS>
240
<CATEGORY>
245
<RELATIONSHIPS BETWEEN THOUGHTS>
250
<PAST RELATIONSHIPS>
220
<DOCUMENT>

Fig.2

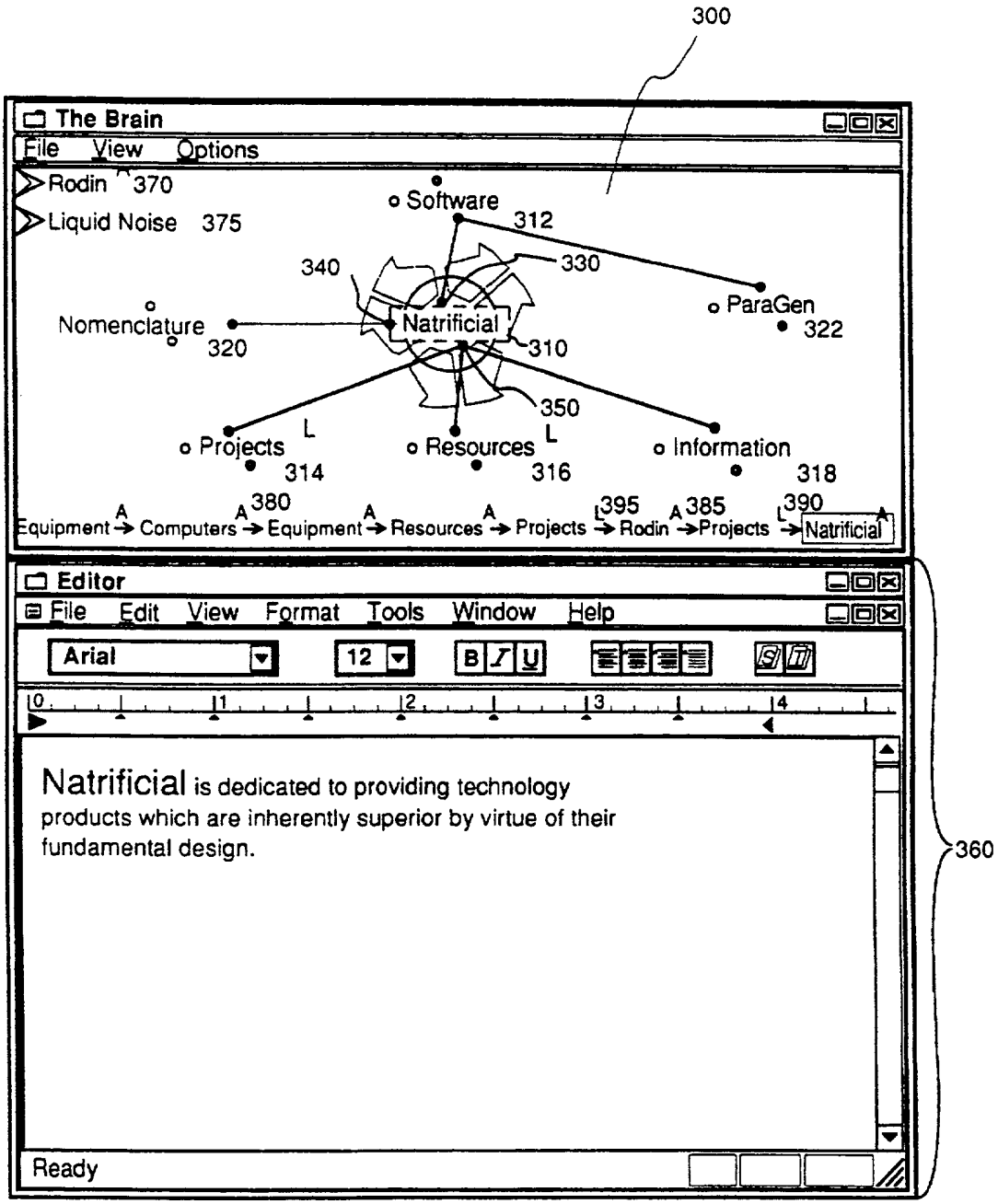

300
The Brain
File View Options
Rodin 370
Liquid Noise 375
Software
312
340
330
Nomenclature
320
Natrificial
310
ParaGen
322
350
Projects
314
Resources
316
Information
318
Equipment → Computers → Equipment → Resources → Projects → Rodin → Projects → Natrificial
380
395
385
390
Editor
File Edit View Format Tools Window Help
Arial
12
B I U
Natrificial is dedicated to providing technology products which are inherently superior by virtue of their fundamental design.
Ready
360

Fig. 3

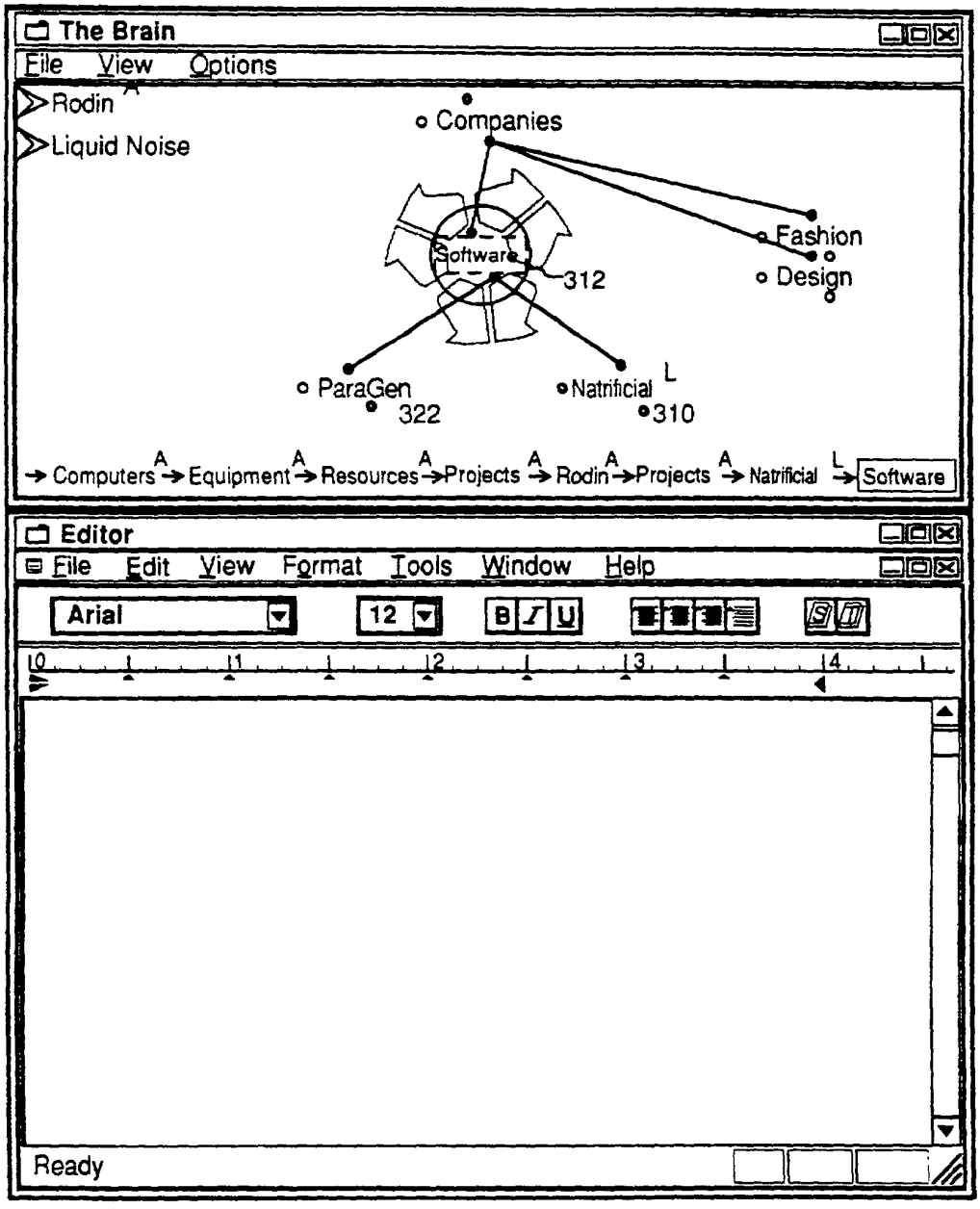

The Brain
File View Options
Rodin
Liquid Noise
Companies
Software
312
Fashion
Design
ParaGen
322
Natrificial
310
Computers A Equipment A Resources A Projects A Rodin A Projects A Natrificial L Software
Editor
File Edit View Format Tools Window Help
Arial
12
B I U
Ready

Fig. 4

Document: 2 of 2

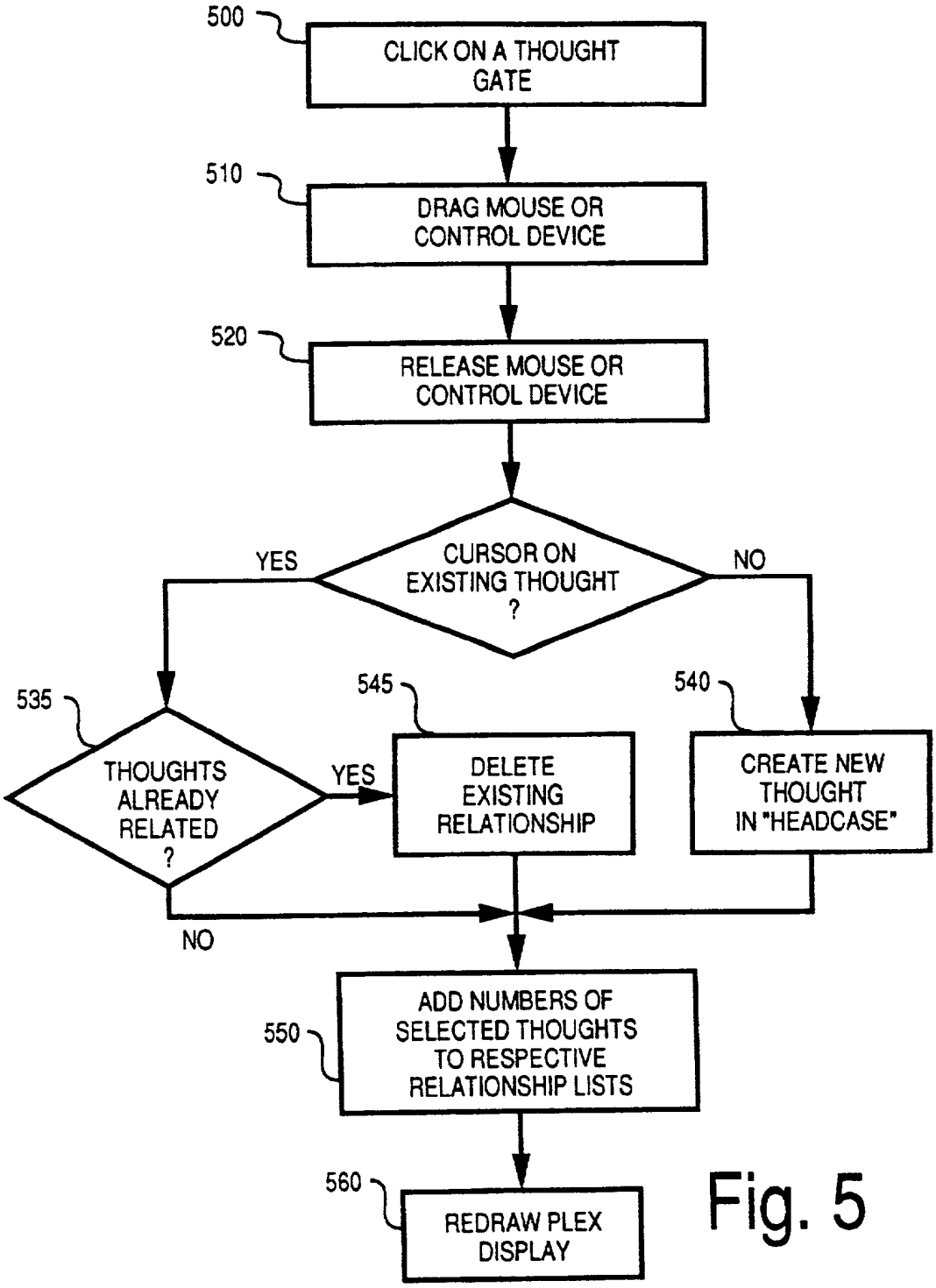


Fig. 5

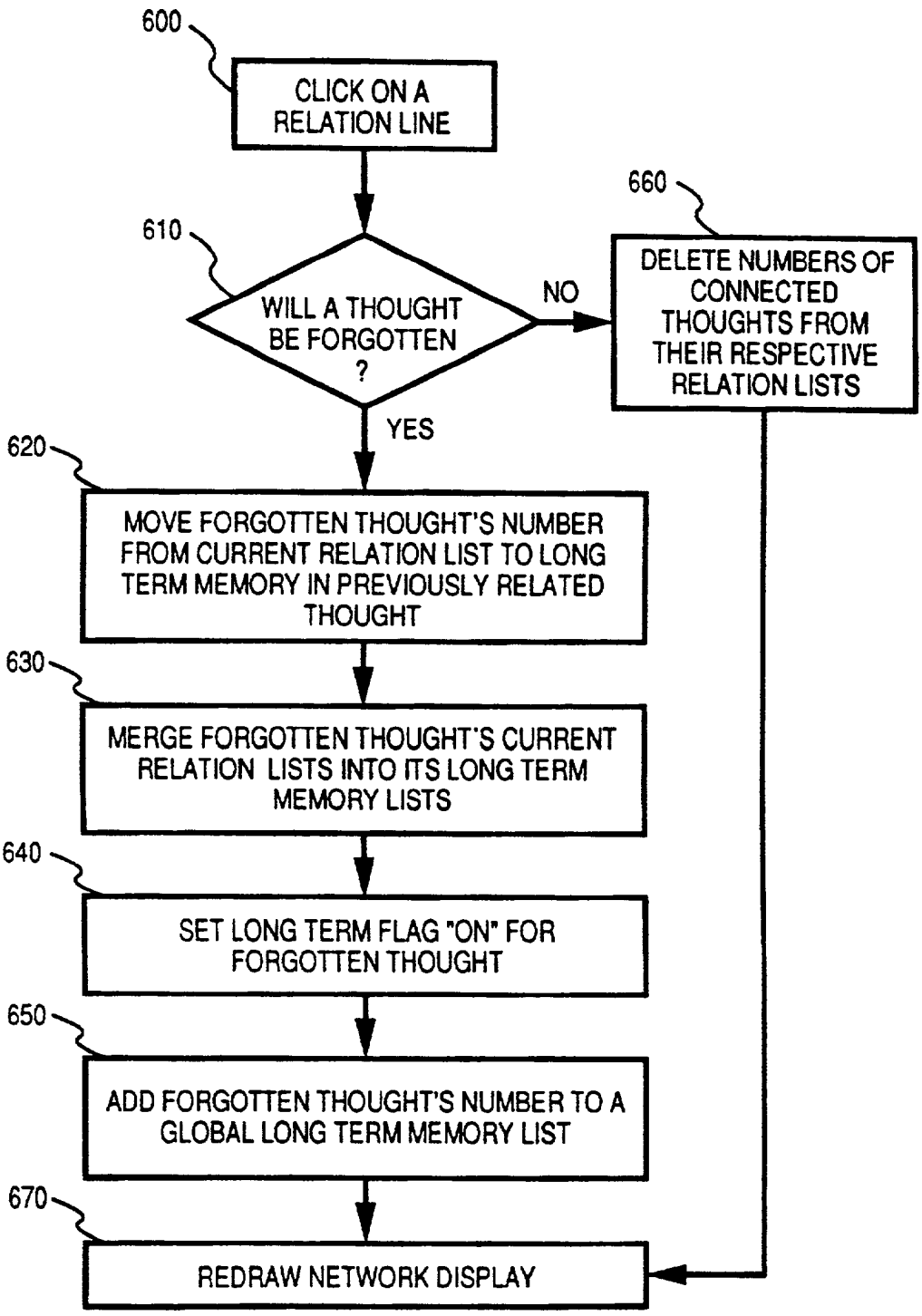

600
CLICK ON A RELATION LINE
610
WILL A THOUGHT BE FORGOTTEN ?
NO
660
DELETE NUMBERS OF CONNECTED THOUGHTS FROM THEIR RESPECTIVE RELATION LISTS
YES
620
MOVE FORGOTTEN THOUGHT'S NUMBER FROM CURRENT RELATION LIST TO LONG TERM MEMORY IN PREVIOUSLY RELATED THOUGHT
630
MERGE FORGOTTEN THOUGHT'S CURRENT RELATION LISTS INTO ITS LONG TERM MEMORY LISTS
640
SET LONG TERM FLAG "ON" FOR FORGOTTEN THOUGHT
650
ADD FORGOTTEN THOUGHT'S NUMBER TO A GLOBAL LONG TERM MEMORY LIST
670
REDRAW NETWORK DISPLAY

Fig. 6

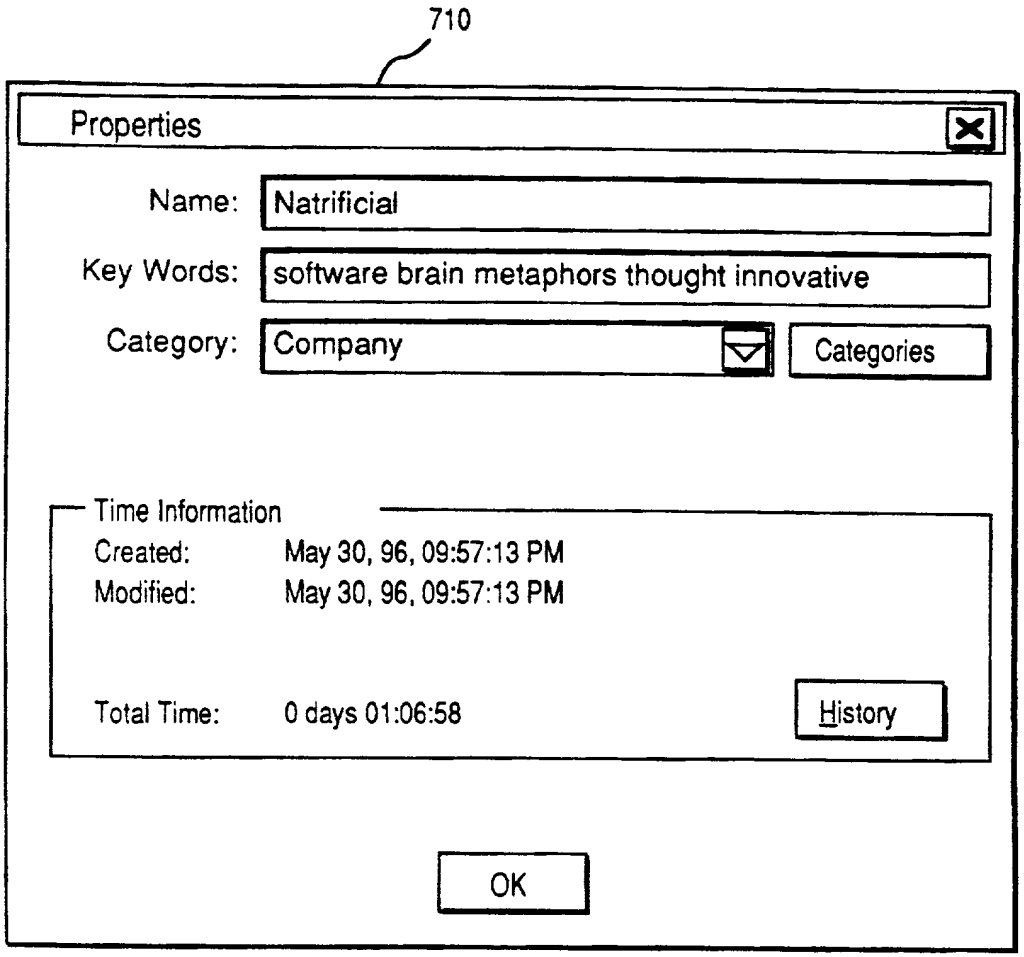

710
Properties
Name:
Natrificial
Key Words:
software brain metaphors thought innovative
Category:
Company
Categories
Time Information
Created:
May 30, 96, 09:57:13 PM
Modified:
May 30, 96, 09:57:13 PM
Total Time:
0 days 01:06:58
History
OK

Fig. 7

Patent Application Publication

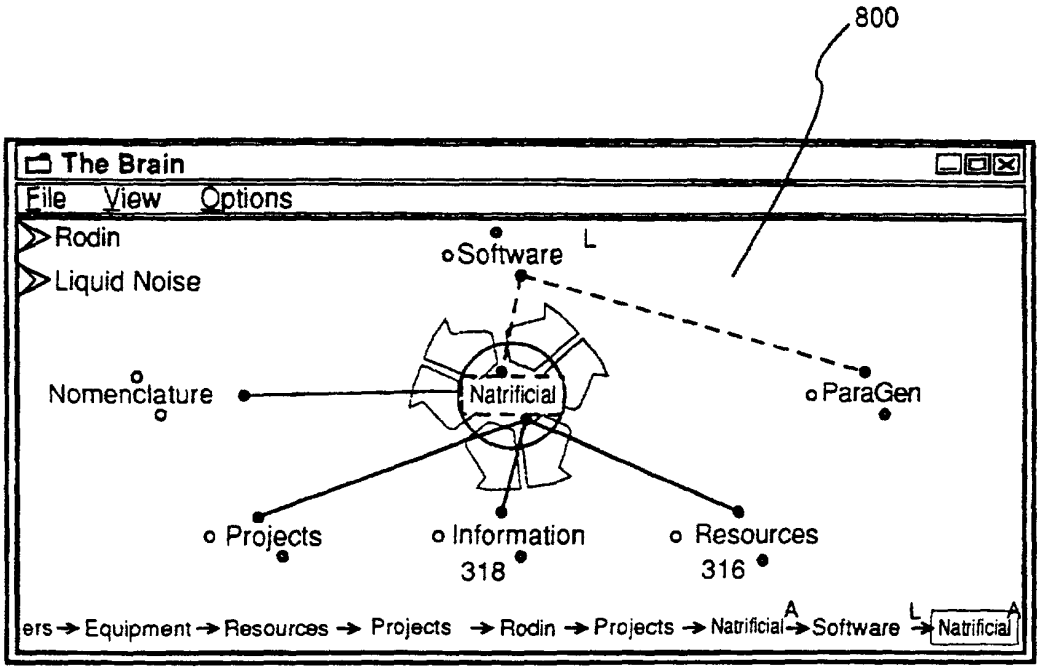


Fig. 8

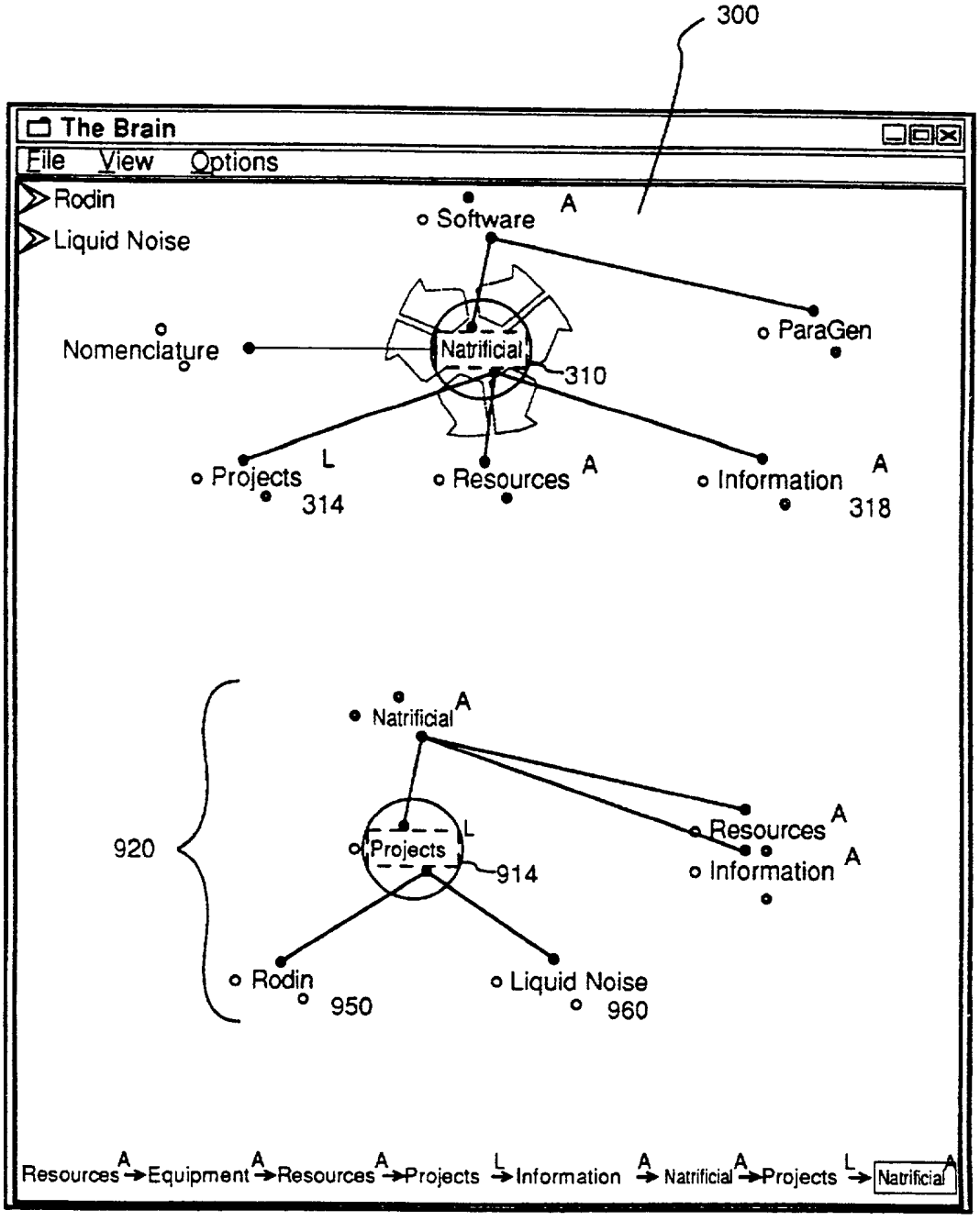

300
The Brain
File View Options
Rodin
Liquid Noise
Software
A
Nomenclature
Natrificial
310
ParaGen
Projects
L
314
Resources
A
Information
A
318
920
Natrificial
A
Projects
L
914
Resources
A
Information
A
Rodin
950
Liquid Noise
960
Resources A→Equipment A→Resources A→Projects L→Information A→Natrificial A→Projects L→Natrificial A

Fig. 9

```
boolean CheckForIsolation(int centralThought, int targetThought)
{
        // this function checks if centralThought is related to targetThought
        // via any of targetThought's relations (not directly)

        // remove centralThought as a direct relation from targetThought
        RemoveRelation(targetThought, centralThought);

        // create an empty thought list to keep track of the search
        intList searchList = CreateEmptyList();

        // start recursive searches on each of targetThought's direct relations
        int relation = GetFirstRelation(targetThought);
        boolean found;
        do {
                found = Search(relation, centralThought, searchList);
                if(found) {
                        // centralThought was found, no need to search any further
                        break;
                }
                // this loop will end when there are no more relations
        } while(relation = GetNextRelation(targetThought);

        // add centralThought back onto target as a relation
        AddRelation(targetThought, centralThought);

        return found;
}
```

Fig. 10a

```
boolean Search(source, dest, searchList)
{
        if(Find(source, searchList)) {
                // source has already been searched
                return FALSE;

        }

        // add source to the searchList
        Add(source, searchList)

        if(source == dest) {
                // this is the destination, we have found it
                return TRUE;
        }

        // recursive searches on each of sources direct relations
        int relation = GetFirstRelation(source);
        boolean found;
        do {
                found = Search(relation, dest, searchList);
                if(found) {
                        // centralThought was found, no need to search any further
                        break;
                }
                // this loop will end when there are no more relations
        } while(relation = GetNextRelation(targetThought);

        return found;
}
```

Fig. 10b

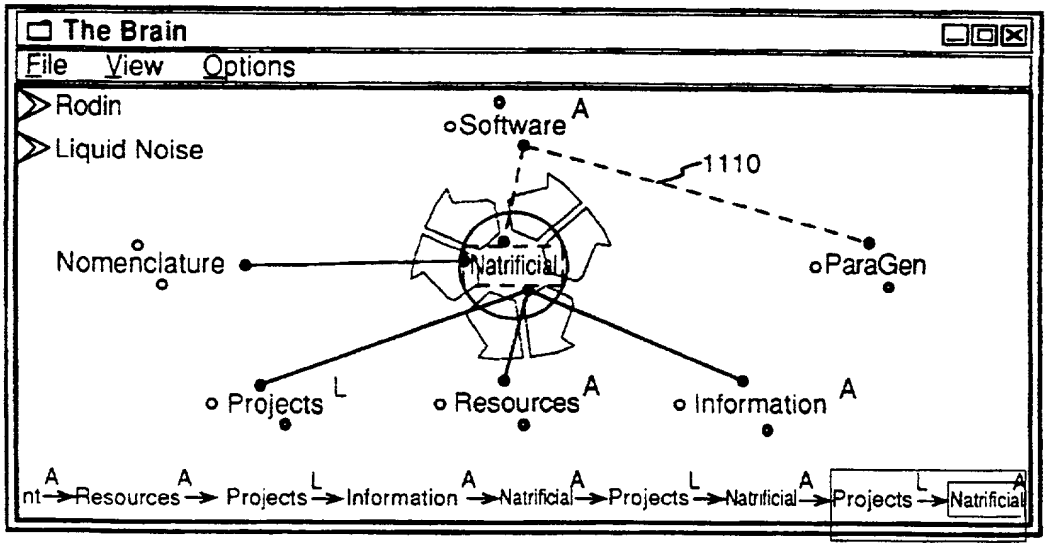

The Brain
File View Options
Rodin
Liquid Noise
Software A
1110
Nomenclature
Natrificial
ParaGen
Projects L
Resources A
Information A
nt A → Resources A → Projects L → Information A → Natrificial A → Projects L → Natrificial A → Projects L → Natrificial A
Create Train of Thought
1120

Fig. 11

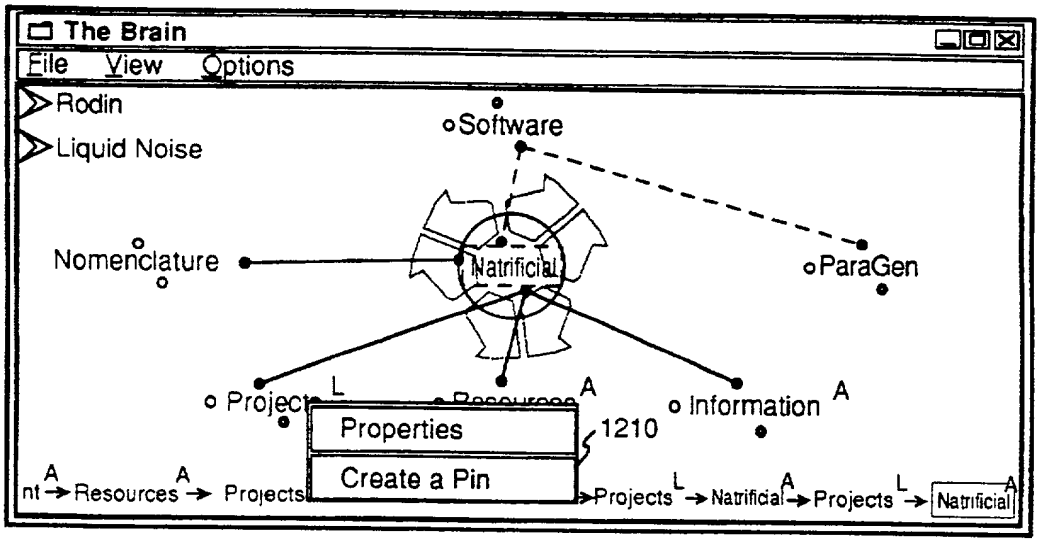

The Brain
File View Options
Rodin
Liquid Noise
Software
Nomenclature
Natrificial
ParaGen
Project
Information
Properties
Create a Pin
1210
Resources
Projects
Natrificial

Fig. 12

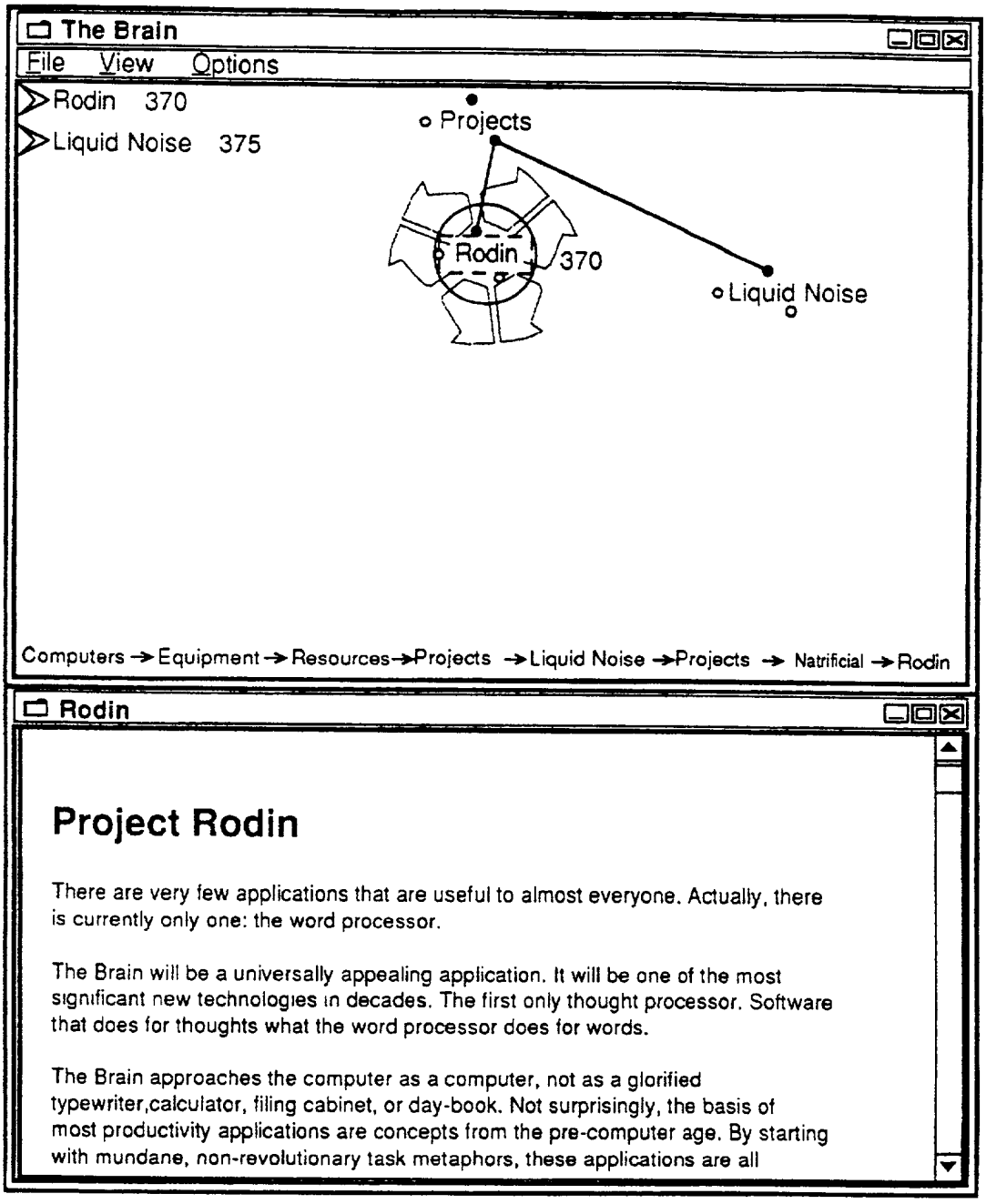

The Brain
File View Options
Rodin 370
Liquid Noise 375
Projects
Rodin
370
Liquid Noise
Computers → Equipment → Resources → Projects → Liquid Noise → Projects → Natrificial → Rodin
Rodin
Project Rodin
There are very few applications that are useful to almost everyone. Actually, there is currently only one: the word processor.
The Brain will be a universally appealing application. It will be one of the most significant new technologies in decades. The first only thought processor. Software that does for thoughts what the word processor does for words.
The Brain approaches the computer as a computer, not as a glorified typewriter,calculator, filing cabinet, or day-book. Not surprisingly, the basis of most productivity applications are concepts from the pre-computer age. By starting with mundane, non-revolutionary task metaphors, these applications are all

Fig. 13

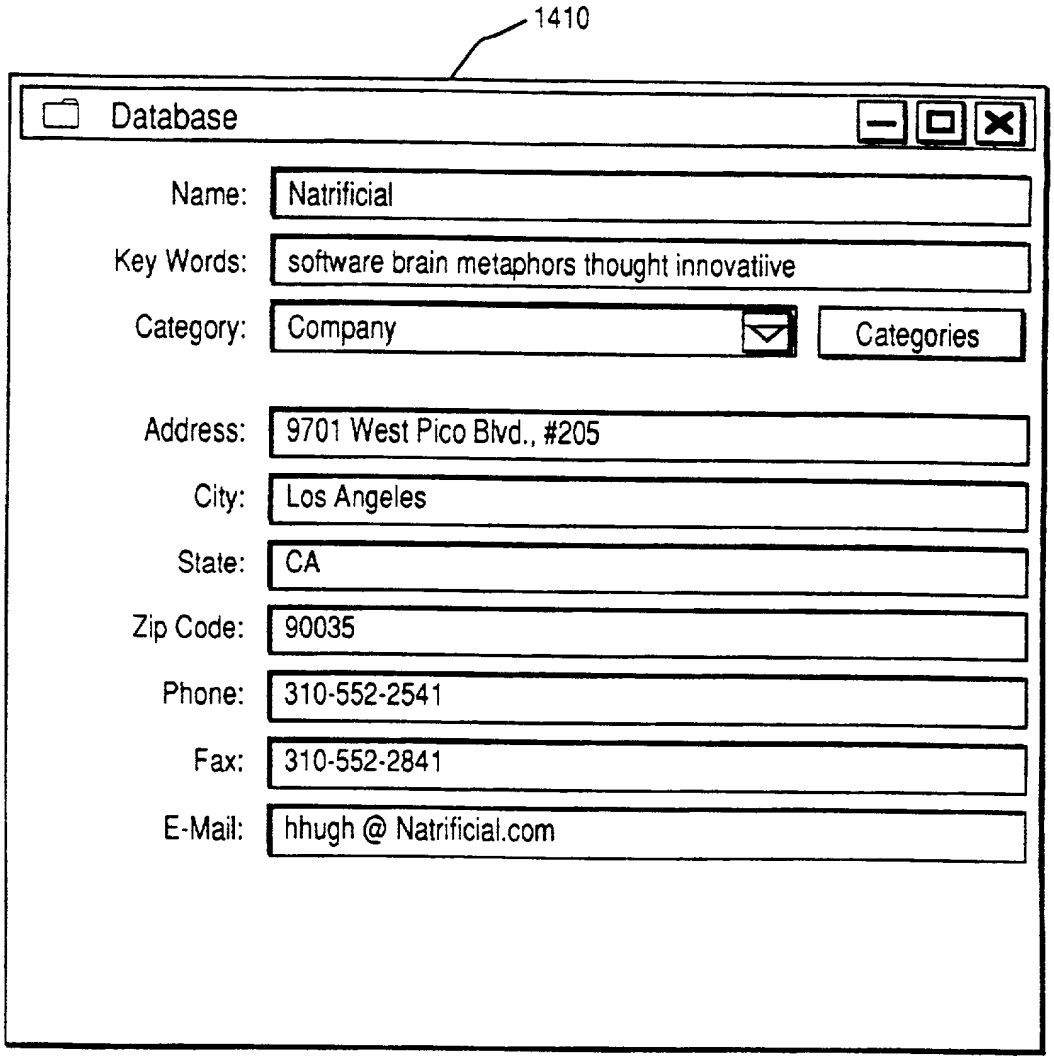

1410
Database
Name:
Natrificial
Key Words:
software brain metaphors thought innovatiive
Category:
Company
Categories
Address:
9701 West Pico Blvd., #205
City:
Los Angeles
State:
CA
Zip Code:
90035
Phone:
310-552-2541
Fax:
310-552-2841
E-Mail:
hhugh @ Natrificial.com

Fig. 14

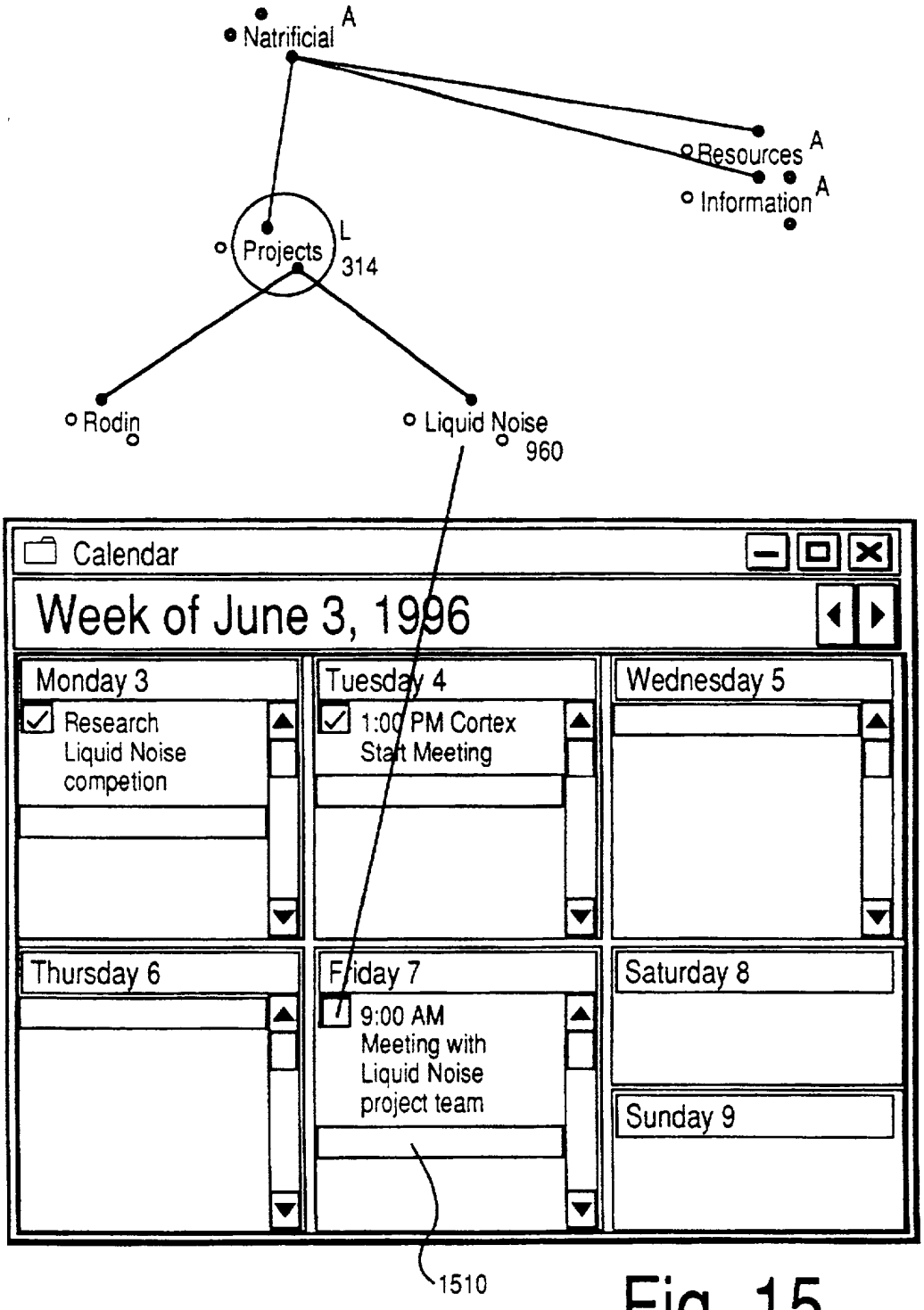

Natrificial
A
Resources
A
Information
A
Projects
L
314
Rodin
Liquid Noise
960
Calendar
Week of June 3, 1996
Monday 3
Research
Liquid Noise
competion
Tuesday 4
1:00 PM Cortex
Staff Meeting
Wednesday 5
Thursday 6
Friday 7
9:00 AM
Meeting with
Liquid Noise
project team
Saturday 8
Sunday 9
1510

Fig. 15

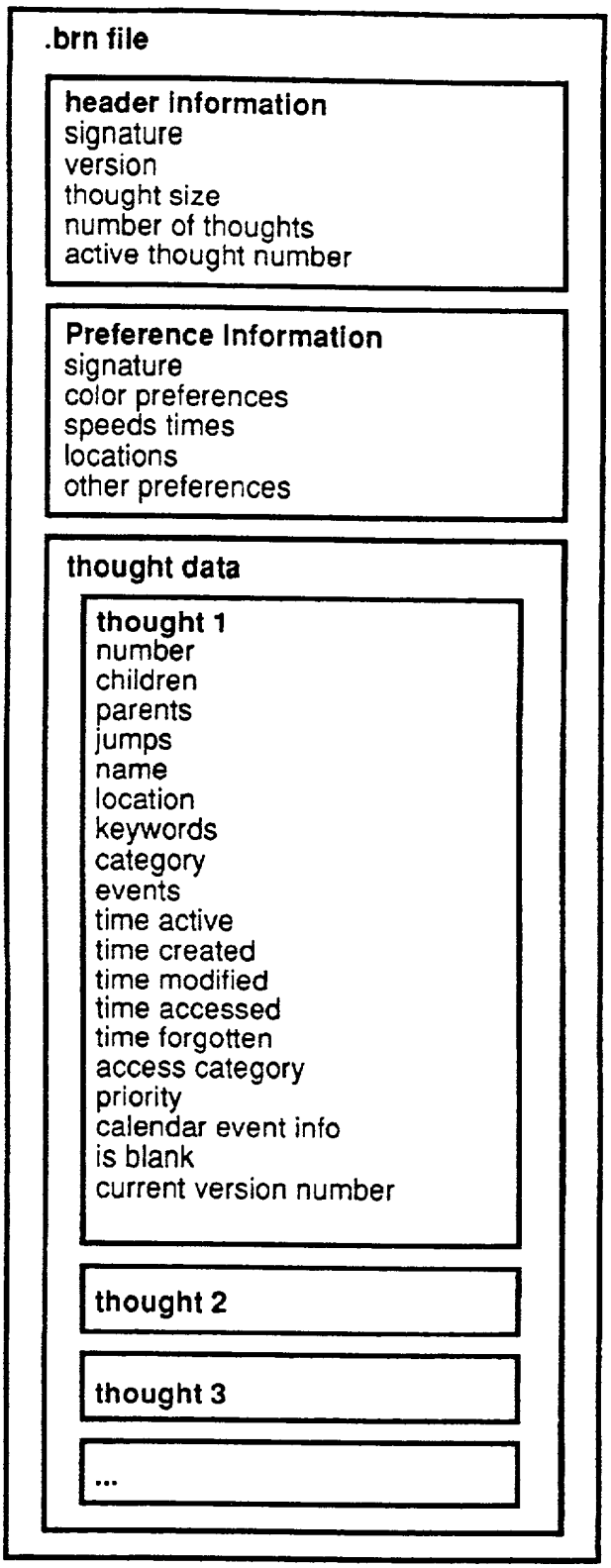

.brn file
header information
signature
version
thought size
number of thoughts
active thought number
Preference Information
signature
color preferences
speeds times
locations
other preferences
thought data
thought 1
number
children
parents
jumps
name
location
keywords
category
events
time active
time created
time modified
time accessed
time forgotten
access category
priority
calendar event info
is blank
current version number
thought 2
thought 3
...

Fig. 16

```
ForgetThought (fNum)
{
  // mark all the children of the selected thought
  list.Clear();
  MarkChildren(fNum, list);
  // unmark the active thought
  list.RemoveThought(activeThought);
  // unmark thoughts with unmarked parents
  lNum = list.GetFirstNum();
  while(lNum !=0)
  {
    if(lNum != fNum)  // don't unmark the selected thought
    {
      pNum = GetFirstThoughtParent(lNum);
      while(pNum != 0)
      {
        if(list.Contains(pNum) == FALSE)
        {
          if(IsThoughtInLongTermMemory(pNum)  == FALSE)
          {
            // unmark all the children of the unmarked parent
            childList.Clear();


            MarkChildren(pNum, childList);
            list.RemoveList(childList);
          }
        }
        pNum = GetNextThoughtParent(lNum);
      }
    }
    lNum = list.GetNextNum();
  }
  // now forget all the thoughts left on the list
  lNum = list.GetFirstNum();
  while(lNum != 0)
  {
    ForgetThought(lNum);
    lNum = list.GetNextNum();
  }
}

RememberThought(rNum)
{
  // mark all the children of the selected thought
  list.Clear();
  MarkChildren(rNum, list);
  // remember all the thoughts on the list
  lNum = list.GetFirstNum();
  while(lNum != 0)
  {
    RememberThought(lNum);
    lNum = list.GetNextNum();
  }
}

MarkChildren(num, list)
{
  list.AddThought(num);
  cNum = GetFirstChild(num);
  while(cNum != 0)
  {
    MarkChildren(cNum, list);
    cNum = GetNextChild(num);
  }
}
```

FIG. 17

TOUCH 25T2000T

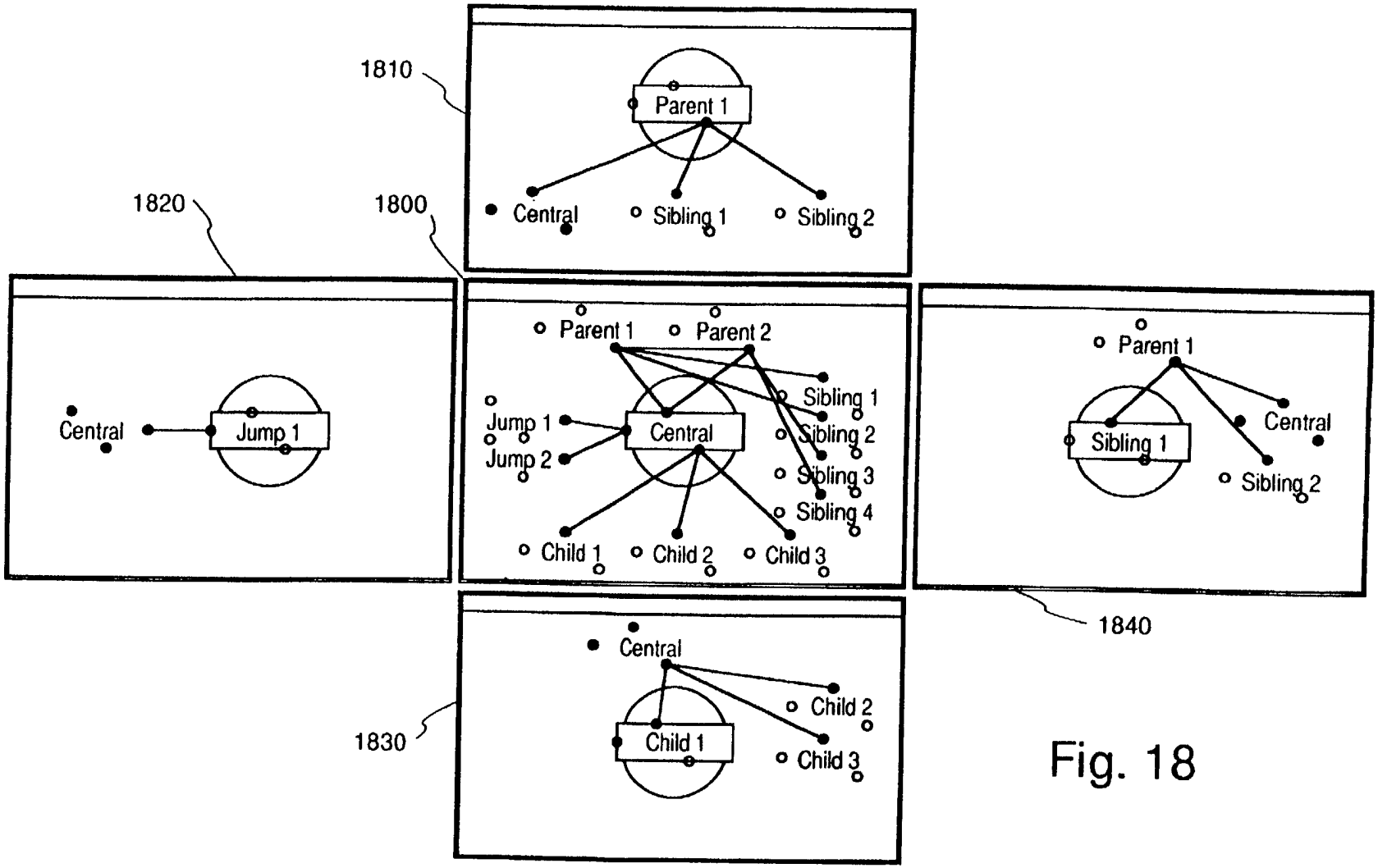


Fig. 18

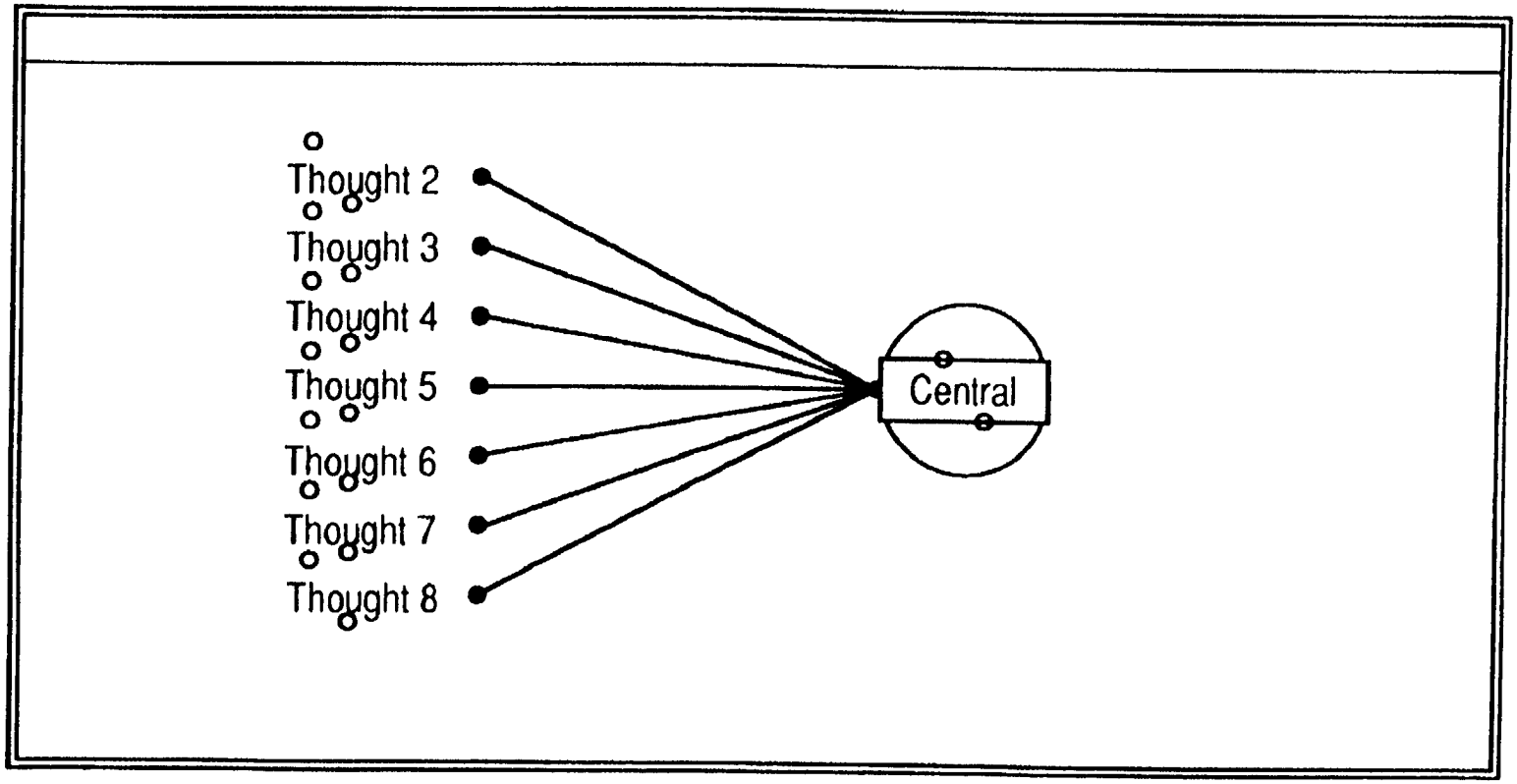

Thought 2
Thought 3
Thought 4
Thought 5
Thought 6
Thought 7
Thought 8
Central

Fig. 19

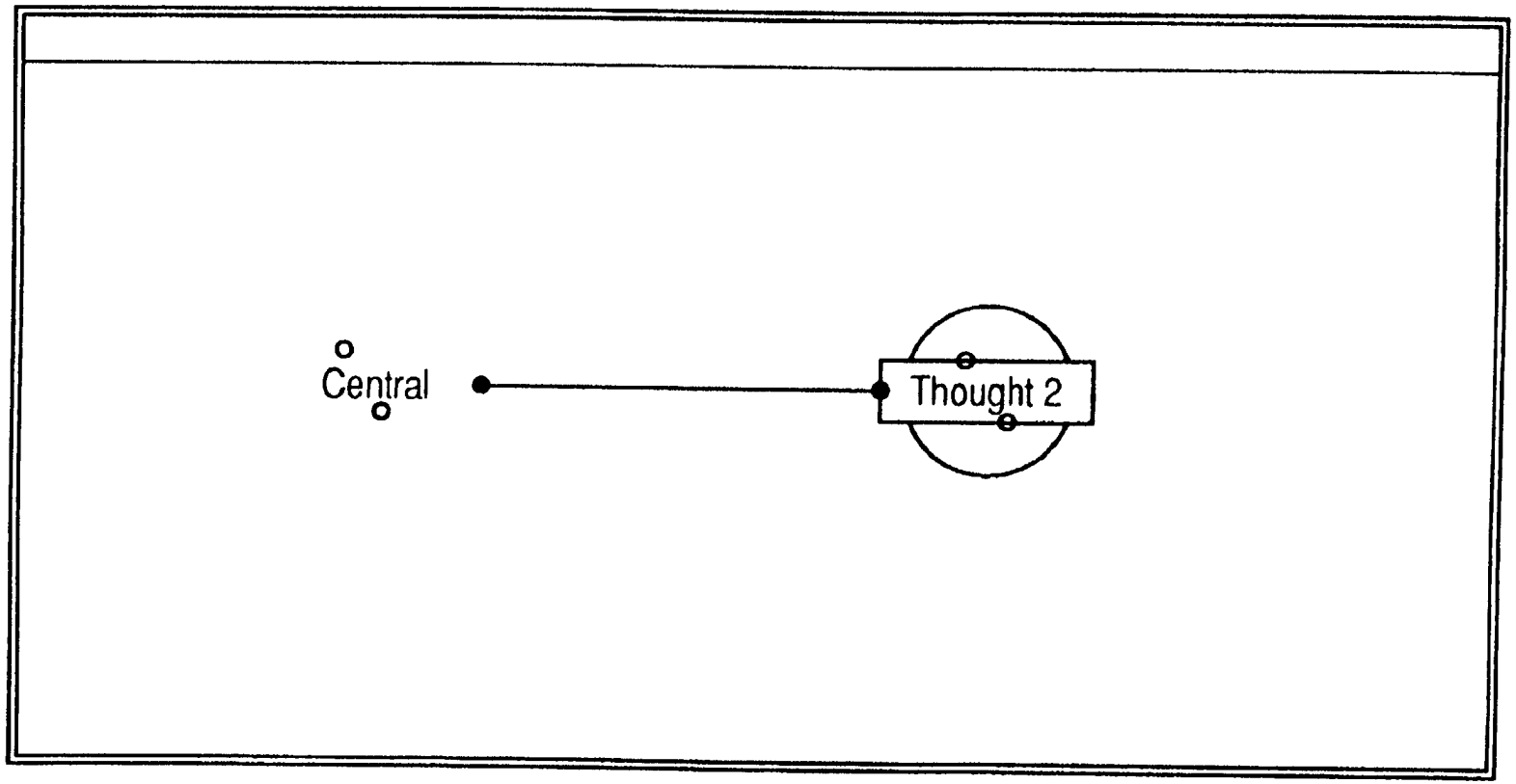


Fig. 20

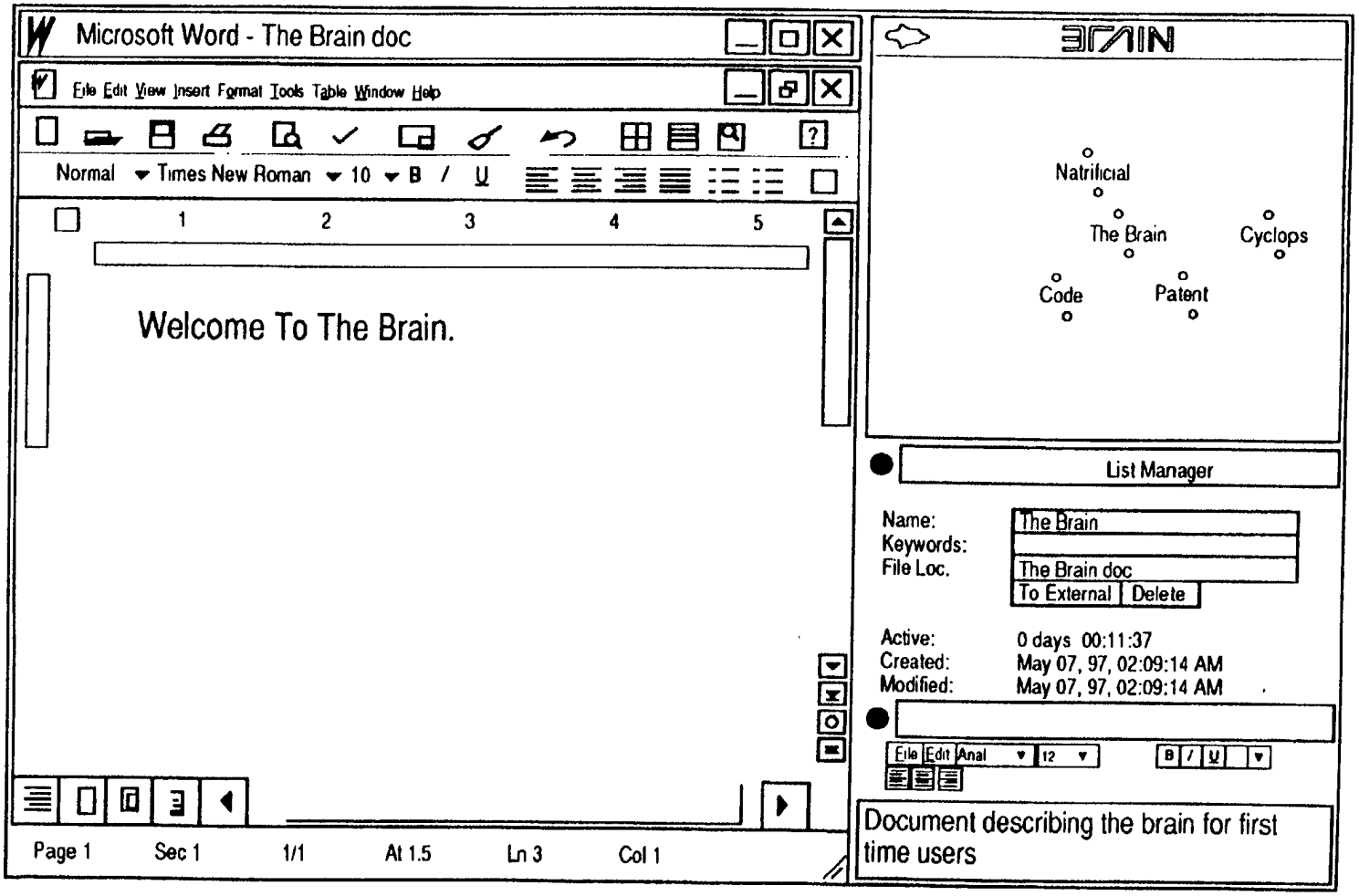

Microsoft Word - The Brain doc
File Edit View Insert Format Tools Table Window Help
Normal Times New Roman 10 B I U
Welcome To The Brain.
Page 1 Sec 1 1/1 At 1.5 Ln 3 Col 1
BRAIN
Natrificial
The Brain
Cyclops
Code
Patent
List Manager
Name: The Brain
Keywords:
File Loc. The Brain doc
To External Delete
Active: 0 days 00:11:37
Created: May 07, 97, 02:09:14 AM
Modified: May 07, 97, 02:09:14 AM
Document describing the brain for first time users

Fig. 21

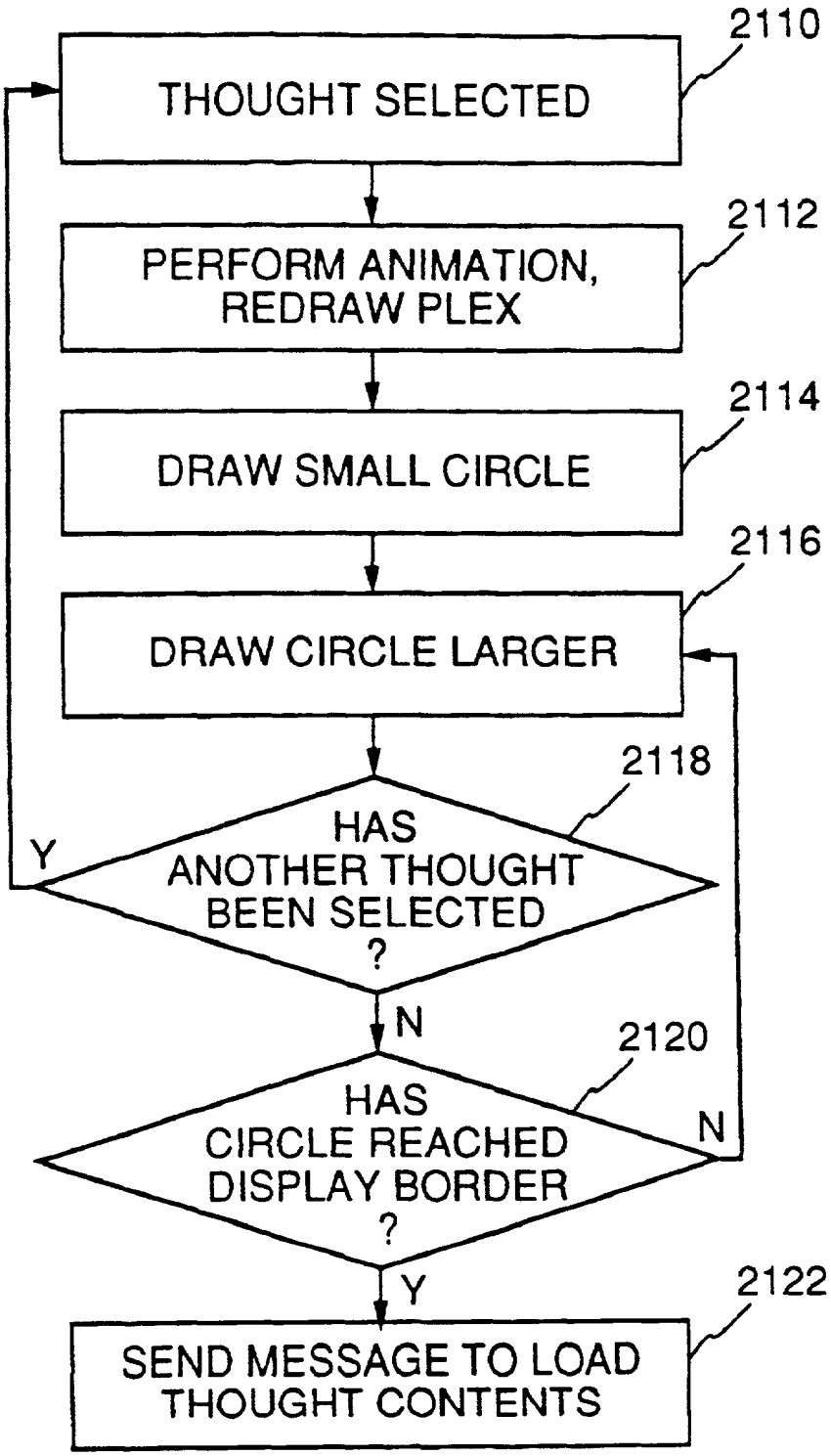

2110
THOUGHT SELECTED
2112
PERFORM ANIMATION, REDRAW PLEX
2114
DRAW SMALL CIRCLE
2116
DRAW CIRCLE LARGER
2118
HAS ANOTHER THOUGHT BEEN SELECTED ?
Y
N
2120
HAS CIRCLE REACHED DISPLAY BORDER ?
N
Y
2122
SEND MESSAGE TO LOAD THOUGHT CONTENTS

Fig. 22

Algorithm for drawing the plex with distant thoughts

1. Create a list of thoughts to be drawn and their screen locations:
    2. Add the central thought to the list.
    3. Add children to the list.
    4. Add parents to the list.
    5. Add jumps to the list.
    6. Add siblings to the list, checking first that they are not already on the list.
    7. Add distants of children to the list, checking first that they are not already on the list.
    8. Add distants of parents to the list, checking first that they are not already on the list.
    9. Add distants of jumps to the list, checking first that they are not already on the list.
    10. Add distants of siblings to the list, checking first that they are not already on the list.
11. Draw the lines that connect each thought:
    12. For each item in the list:
        13. Get each item in the list:
            14. If the two items are related, draw lines between them from and to the appropriate gates.
15. Draw the distant thoughts:
    16. For each item in the list:
        17. If it is a distant thought, draw it.
18. Draw the other thoughts:
    19. For each item in the list:
        20. If it is not a distant thought, draw it.

Fig. 23

4,605,294 - 4,605,294

```
// the non recursive method for searching thoughts
// tries to find a route from nSrc to nDest other than a direct relation
// returns TRUE if found
boolean Search(int nSrc, int nDest)
{
     //create the lists
     ThoughtList posList;     //list of thoughts that possibly connect
     ThoughtList notList;     //list of thoughts that do not connect
     //empty the lists
     posList.Initialize();
     notList.Initialize();

     //add the source to the not list since we cannot go directly
     to the destination,
     notList.Add(nSrc);

     //since we cannot go directly to the destination,
     //add all relates except the destination to the possible list
     Thought src(nSrc);
     for(int n = 0;;n++)
     {
          int nRel = src.GetRelate(n);
          if(!nRel)
          {
               //no more relations, done
               break;
          }
          if(nRel != nDest)
          {
               //add it to the possibly connect list
               posList.Add(nRel);
          }
     }

     while(TRUE)
     {
          //check the first possibility
          int nTest = posList.GetFirst();
          if(!nTest)
          {
               //nothing on the list, done
               break;
          }
          Thought test(nTest);
          if(test.IsRelated(nDest))
          {
               //this one is related to the destination, we're done
               return TRUE;
          }
          // does not connect, add it to the does not connect list
          notList.Add(nTest);
          // add all related thoughts except those already checked to
          possible list for
          (int n = 0;; n++)
          {
               int nRel = test.GetRelate(n);
               if(!nRel)
               {
                              //no more relations, done
                              break;
               }
               if(!notList.Exists(nRel))
               {
                              //not checked yet, add to possible list
                              posList.Add(rel);
               }
          }
          //remove this one from the possible list
          posList.Remove(nTest);
     }
     //we've checked everything there is no other way to get from
     nSrc to nDest
     return FALSE;
}
```

Fig. 24

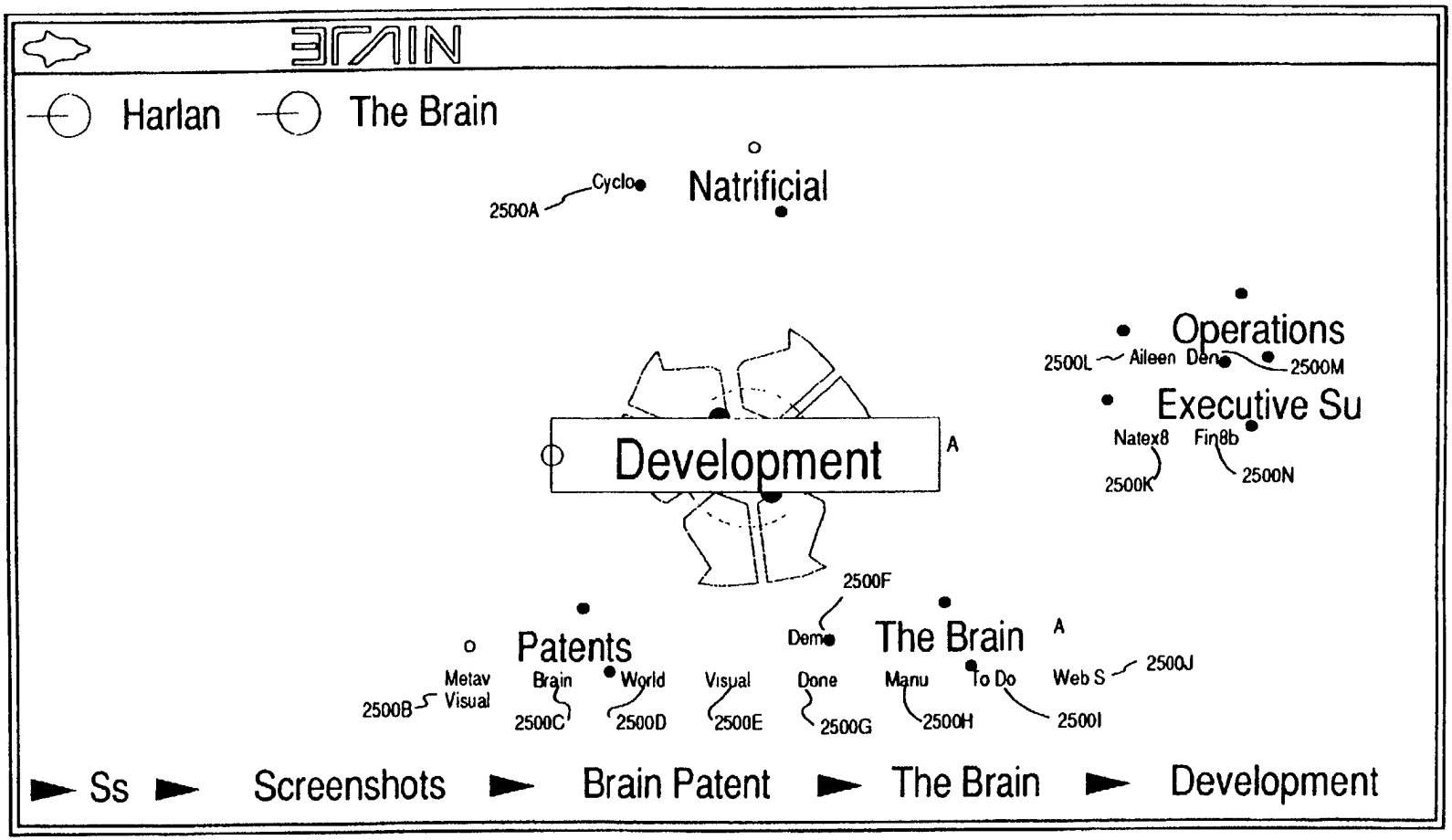

Harlan
The Brain
Natrificial
Cyclo
2500A
Development
A
Operations
2500L
Aileen
Den
2500M
Executive Su
Natex8
Fin8b
2500K
2500N
2500F
Patents
Demo
The Brain
A
Metav
Visual
2500B
Brain
World
Visual
Done
Manu
To Do
Web S
2500J
2500C
2500D
2500E
2500G
2500H
2500I
Ss
Screenshots
Brain Patent
The Brain
Development

Fig. 25

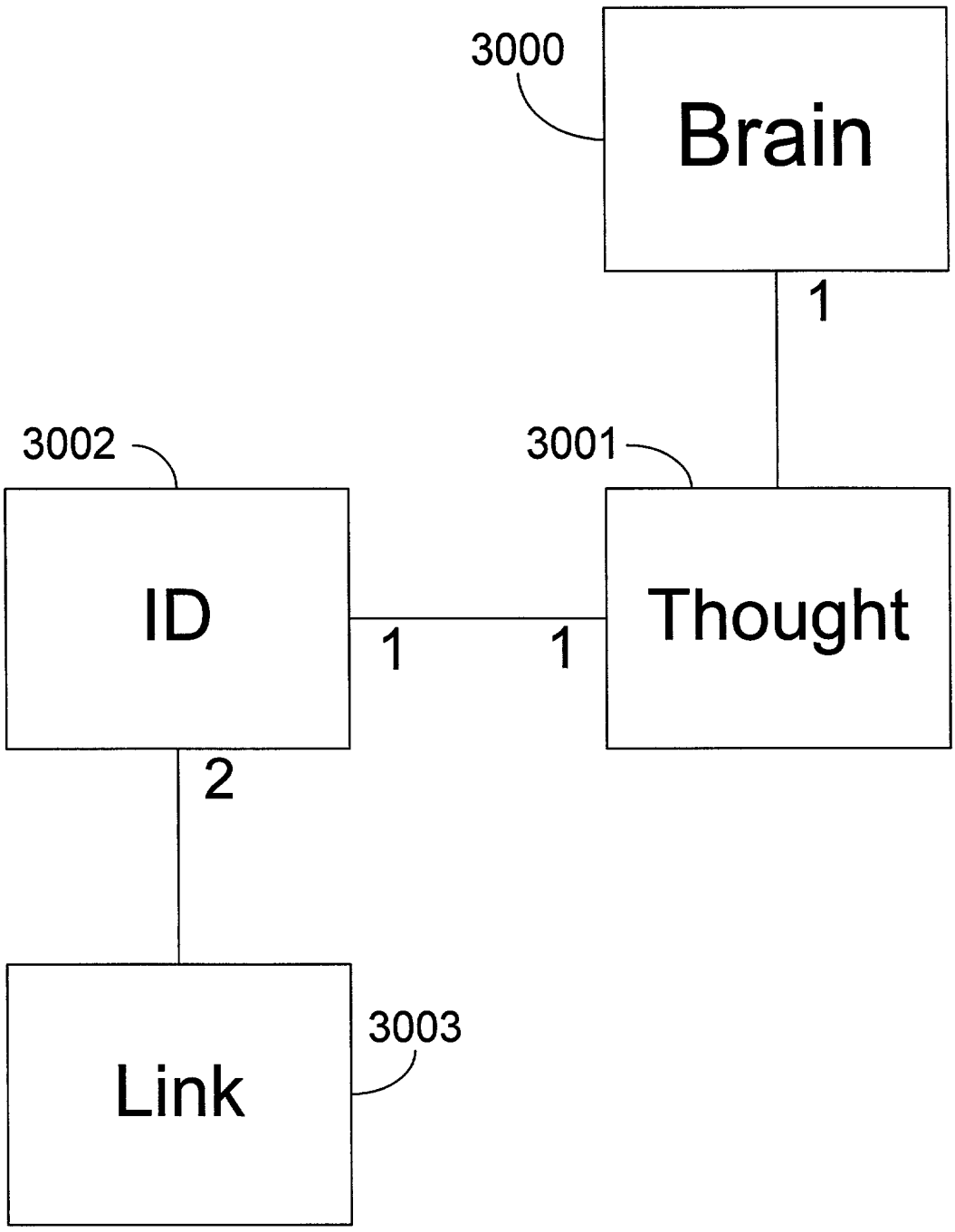


FIG. 26

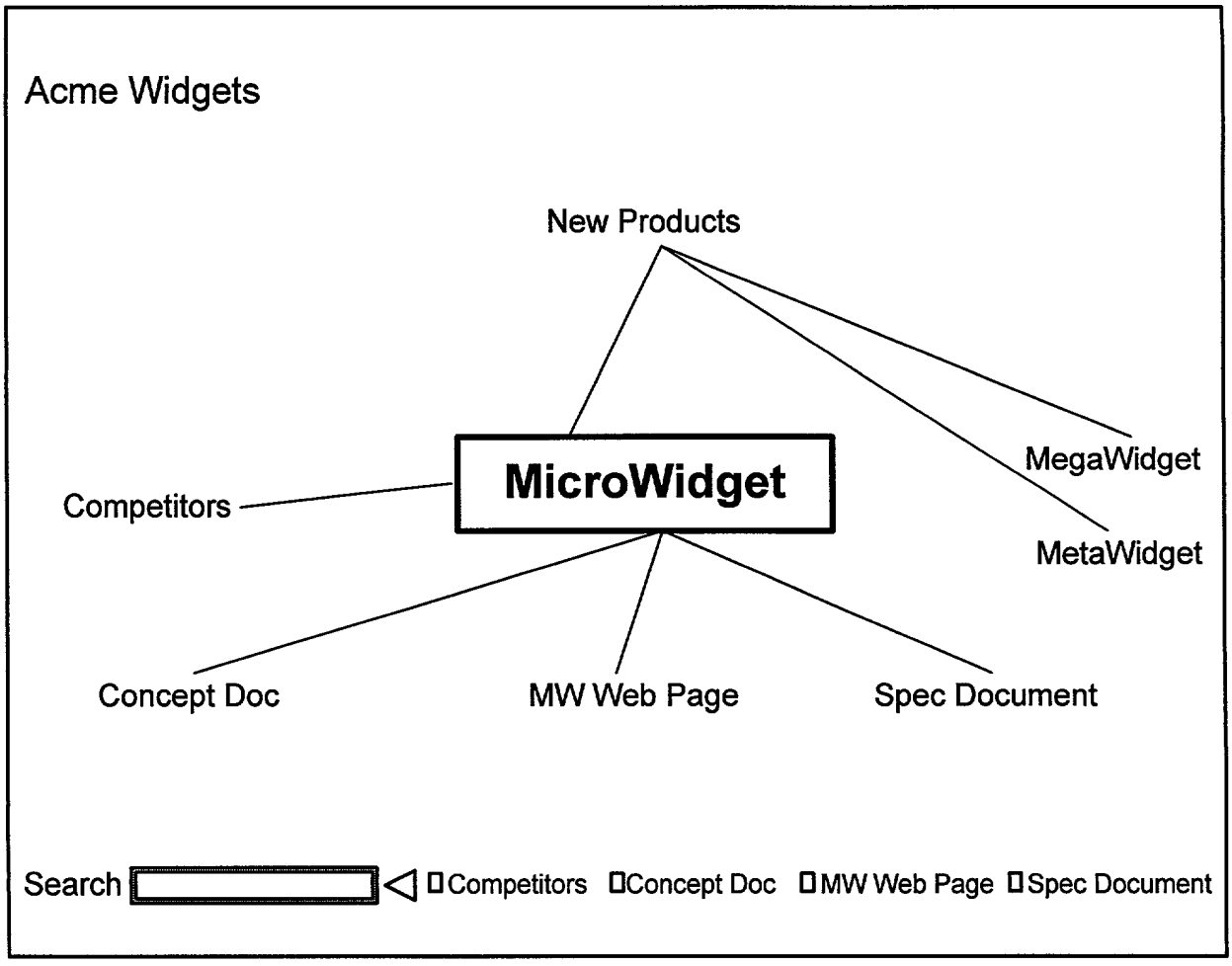
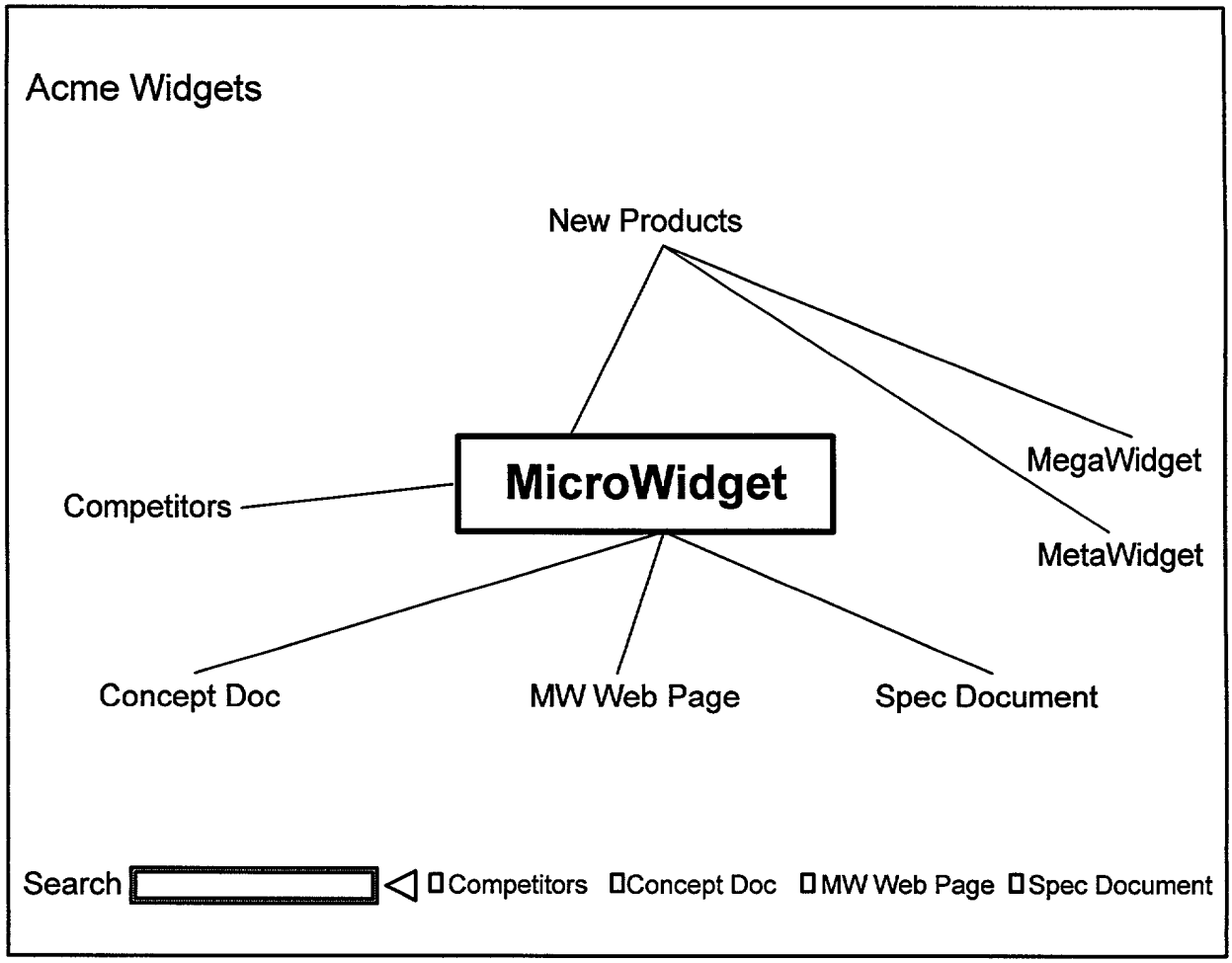

Acme Widgets
New Products
MegaWidget
MetaWidget
Competitors
MicroWidget
Concept Doc
MW Web Page
Spec Document
Search
Competitors
Concept Doc
MW Web Page
Spec Document
Select
Thoughts
Thoughts
Links
where

FIG. 27


Acme Widgets
New Products
MicroWidget
Competitors
MegaWidget
MetaWidget
Concept Doc
MW Web Page
Spec Document
Search
Competitors
Concept Doc
MW Web Page
Spec Document

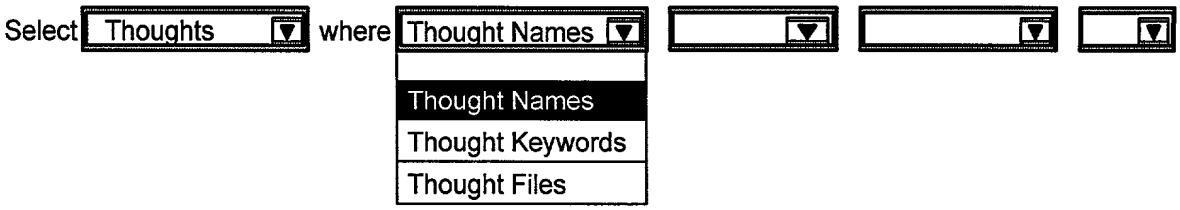

Select
Thoughts
where
Thought Names
Thought Names
Thought Keywords
Thought Files

FIG. 28

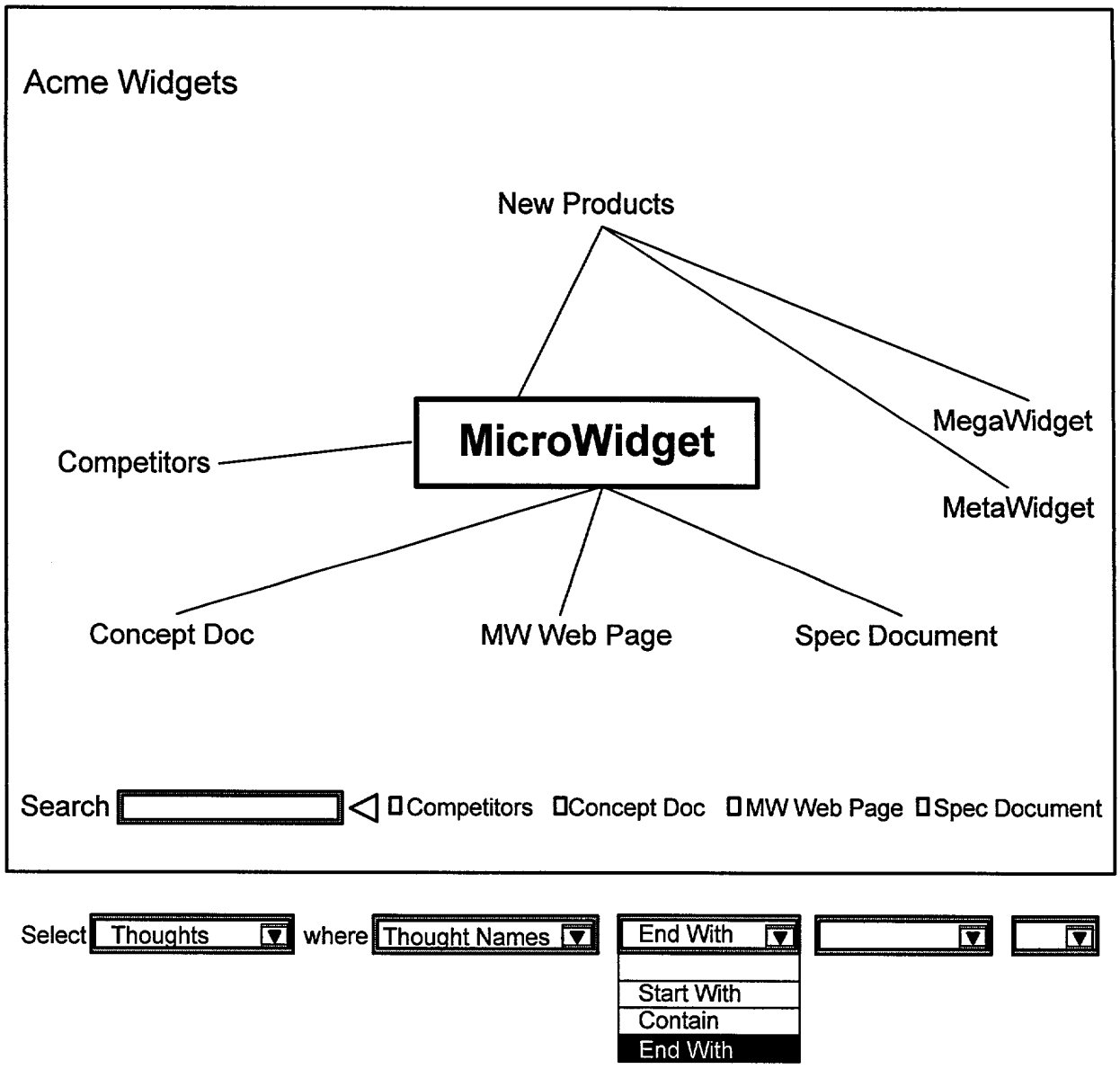

Acme Widgets
New Products
MicroWidget
Competitors
MegaWidget
MetaWidget
Concept Doc
MW Web Page
Spec Document
Search
Competitors
Concept Doc
MW Web Page
Spec Document
Select
Thoughts
where
Thought Names
End With
Start With
Contain
End With

FIG. 29

Patent Application Publication

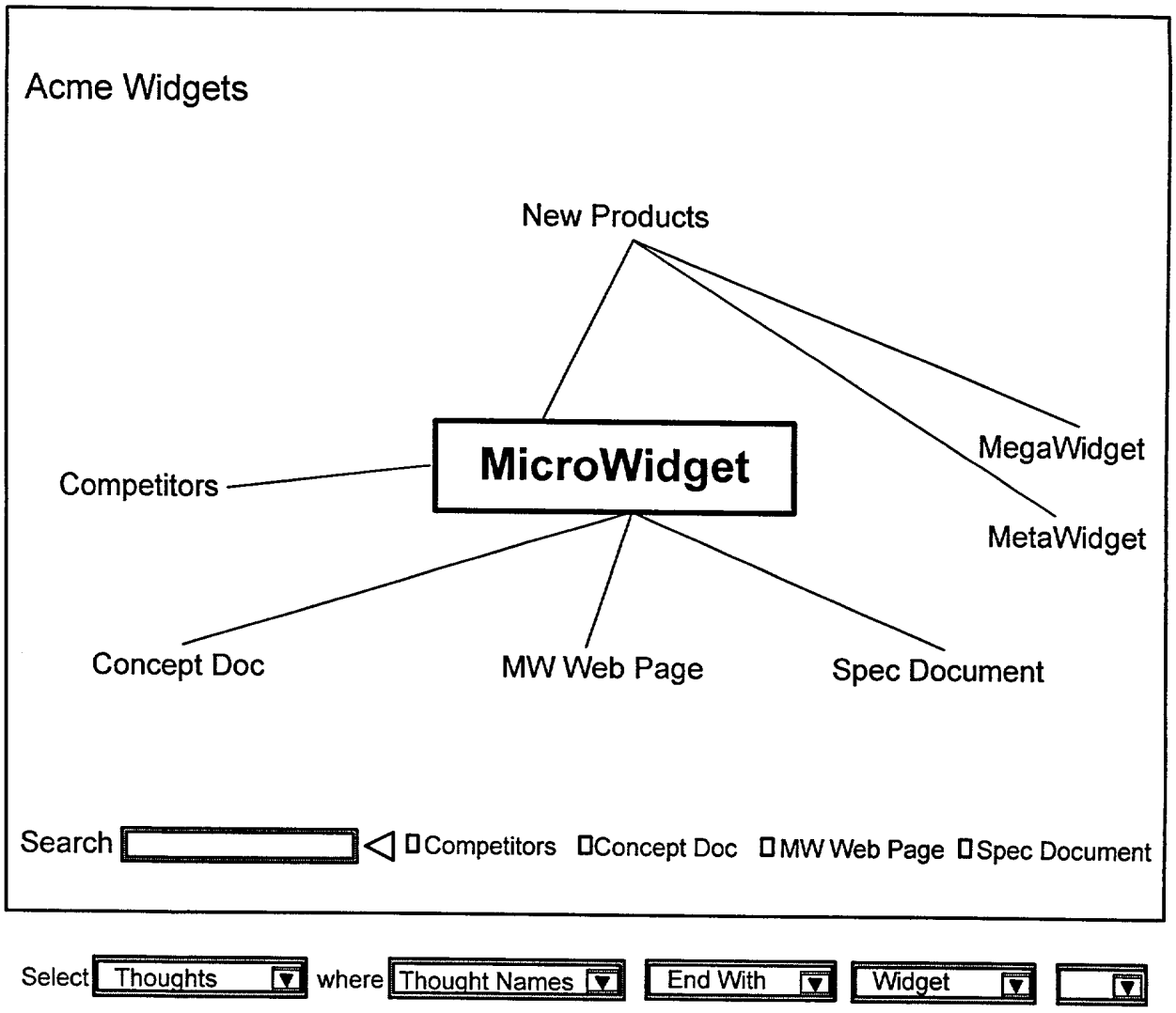


FIG. 30

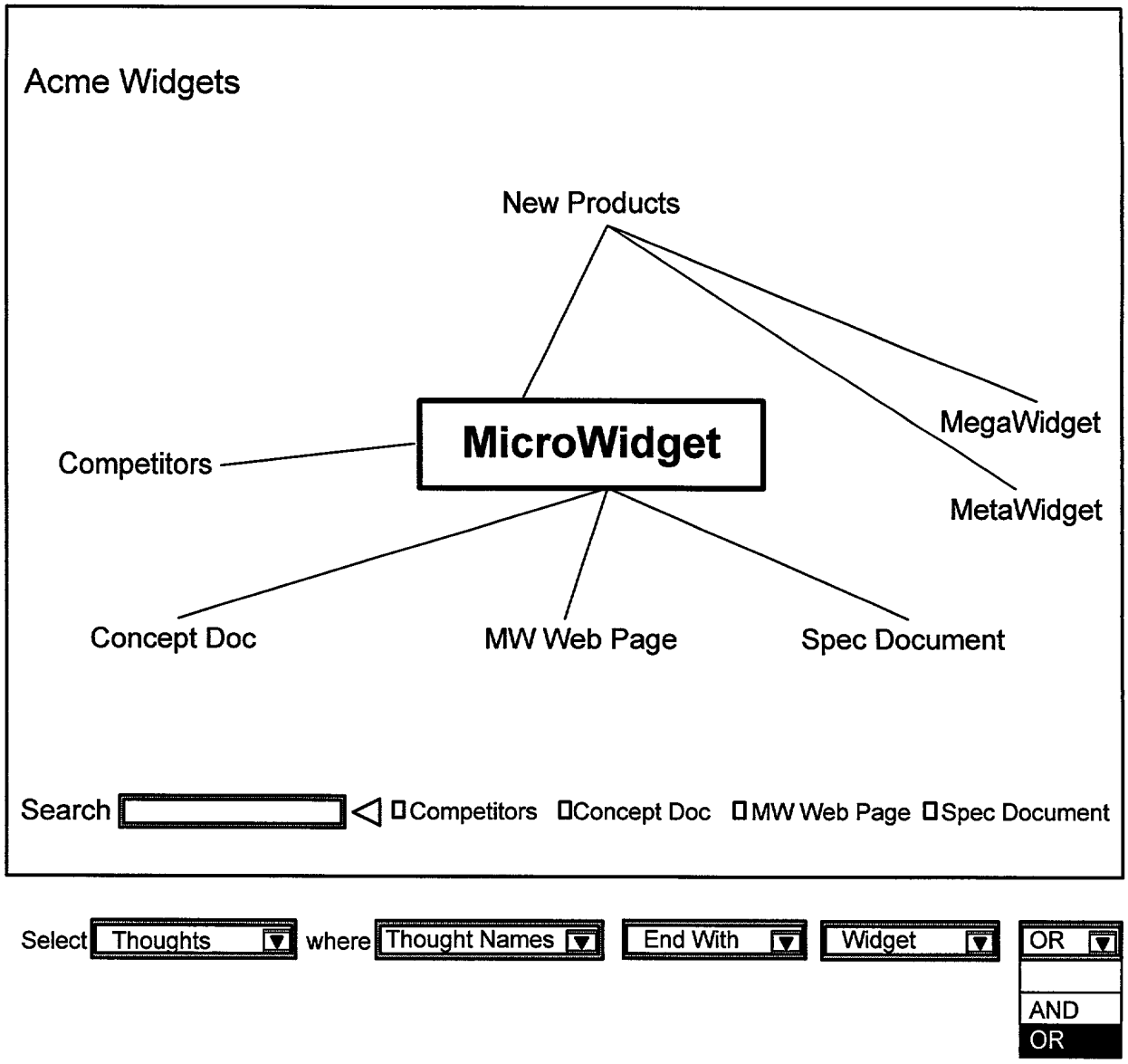

Acme Widgets
New Products
MicroWidget
Competitors
MegaWidget
MetaWidget
Concept Doc
MW Web Page
Spec Document
Search
Competitors
Concept Doc
MW Web Page
Spec Document
Select
Thoughts
where
Thought Names
End With
Widget
OR
AND
OR

FIG. 31

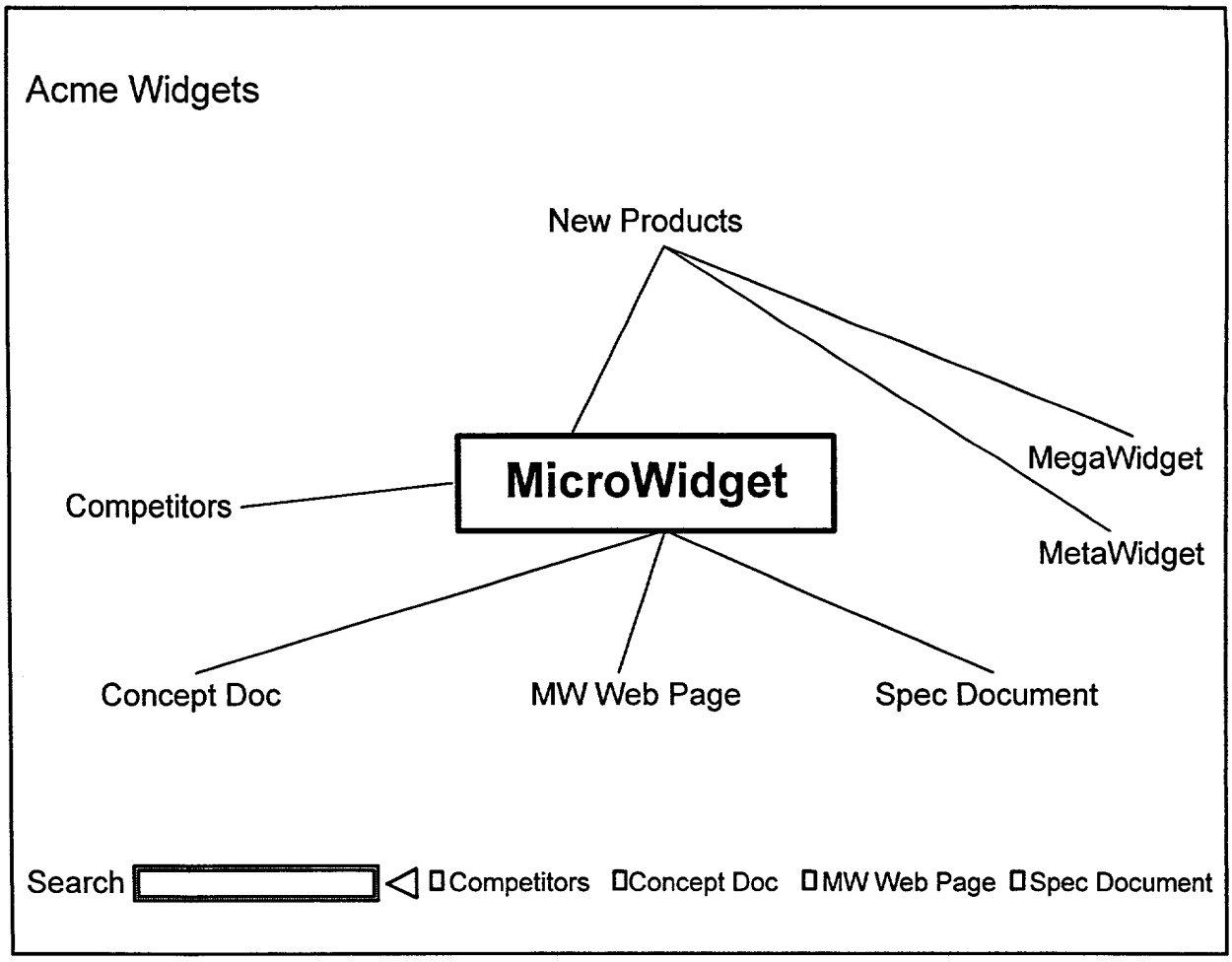

Acme Widgets
New Products
MegaWidget
MetaWidget
Competitors
MicroWidget
Concept Doc
MW Web Page
Spec Document
Search
Competitors
Concept Doc
MW Web Page
Spec Document
Select Thoughts where Thought Names End With Widget OR
Select Thoughts where Thought Keywords Contain Widget

FIG. 32

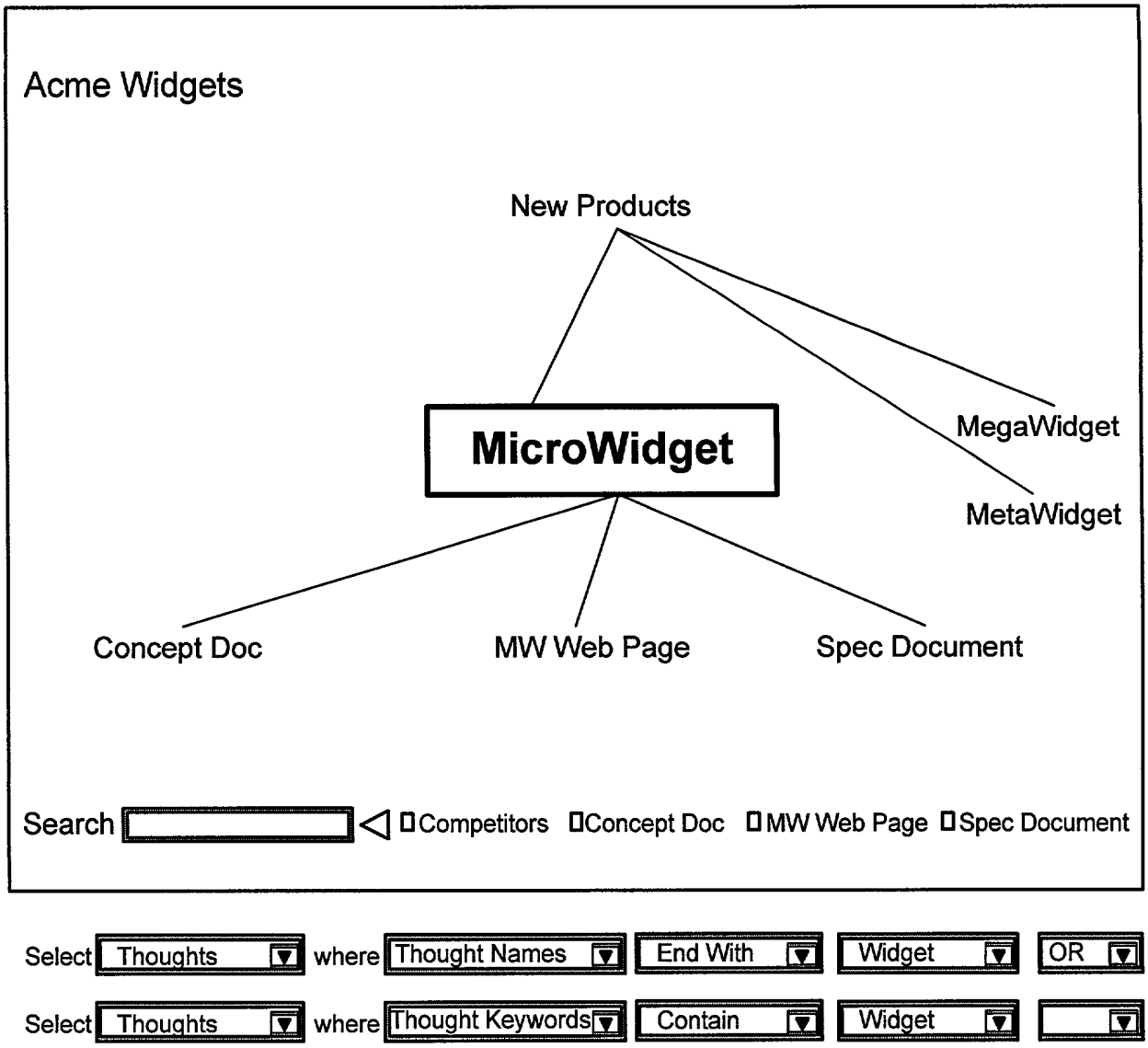

Acme Widgets
New Products
MicroWidget
MegaWidget
MetaWidget
Concept Doc
MW Web Page
Spec Document
Search
Competitors
Concept Doc
MW Web Page
Spec Document
Select Thoughts where Thought Names End With Widget OR
Select Thoughts where Thought Keywords Contain Widget

FIG. 33

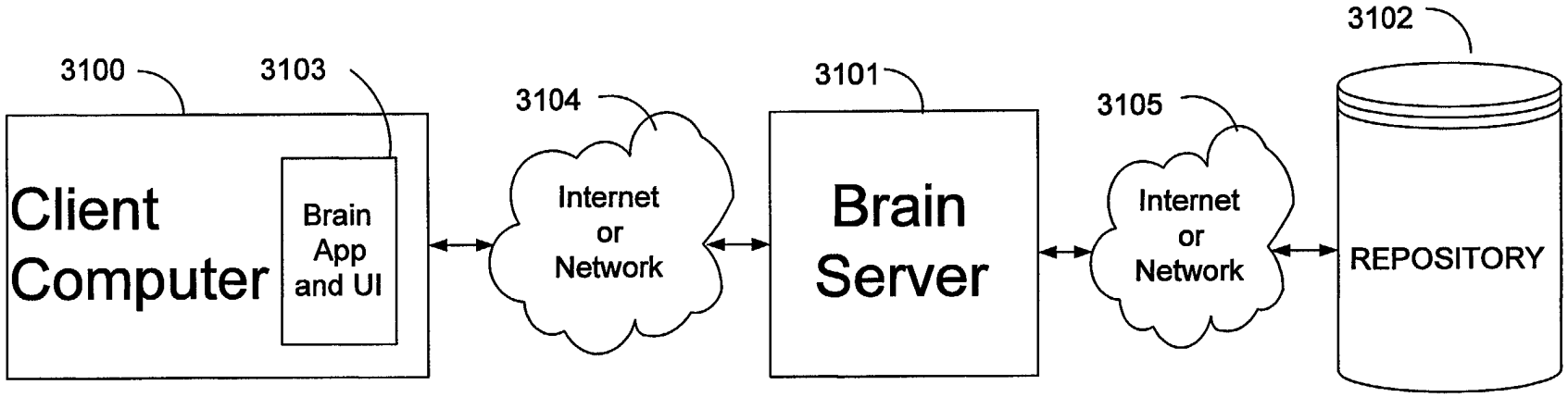
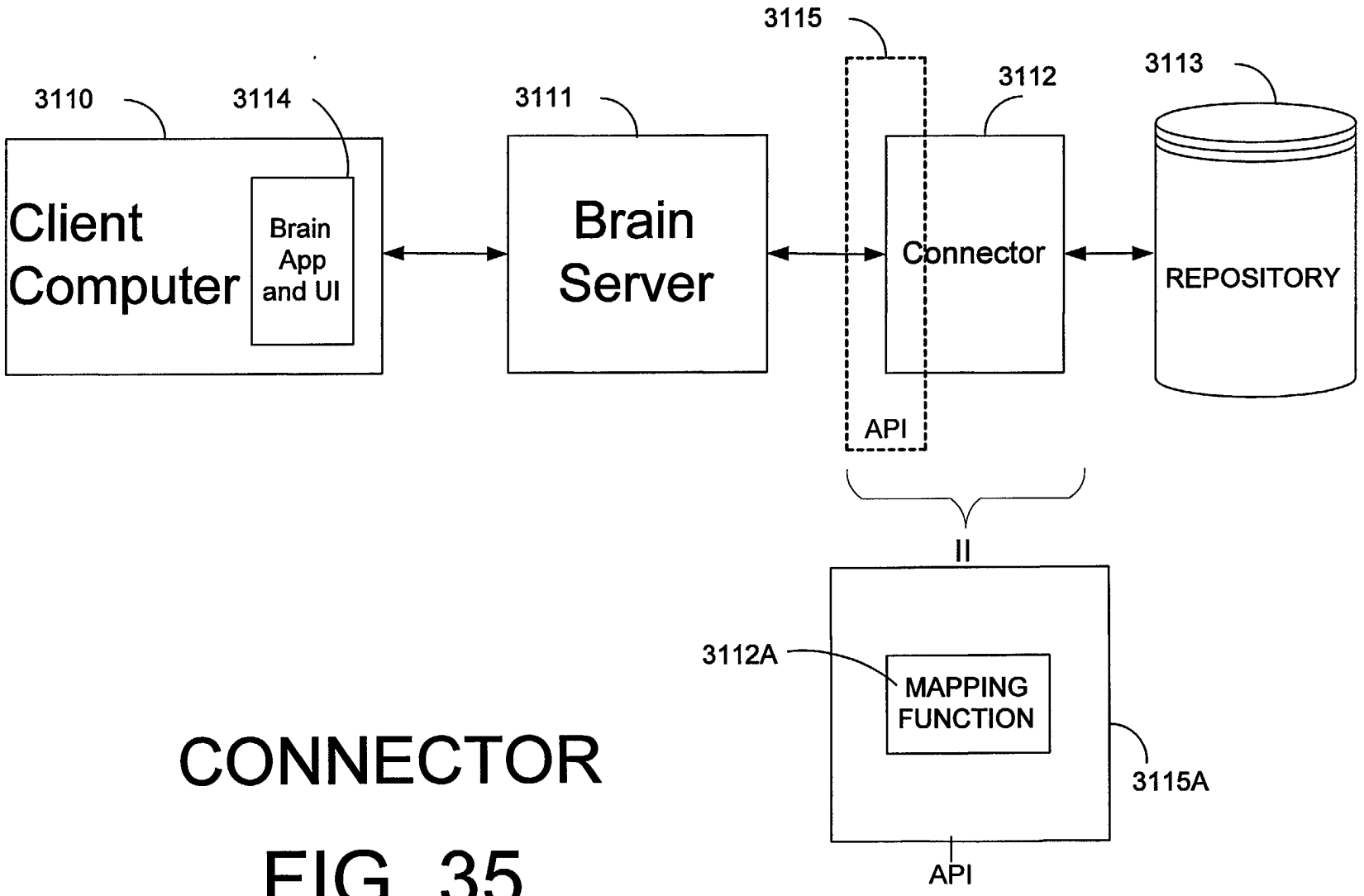

3100
3103
Client Computer
Brain App and UI
3104
Internet or Network
3101
Brain Server
3105
Internet or Network
3102
REPOSITORY

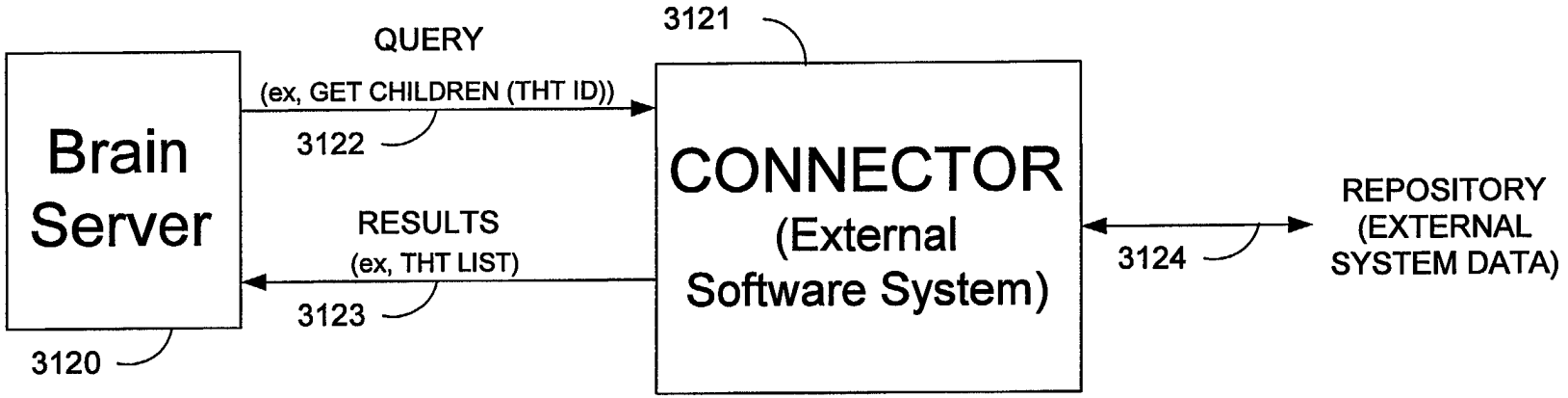
FIG. 34



CONNECTOR

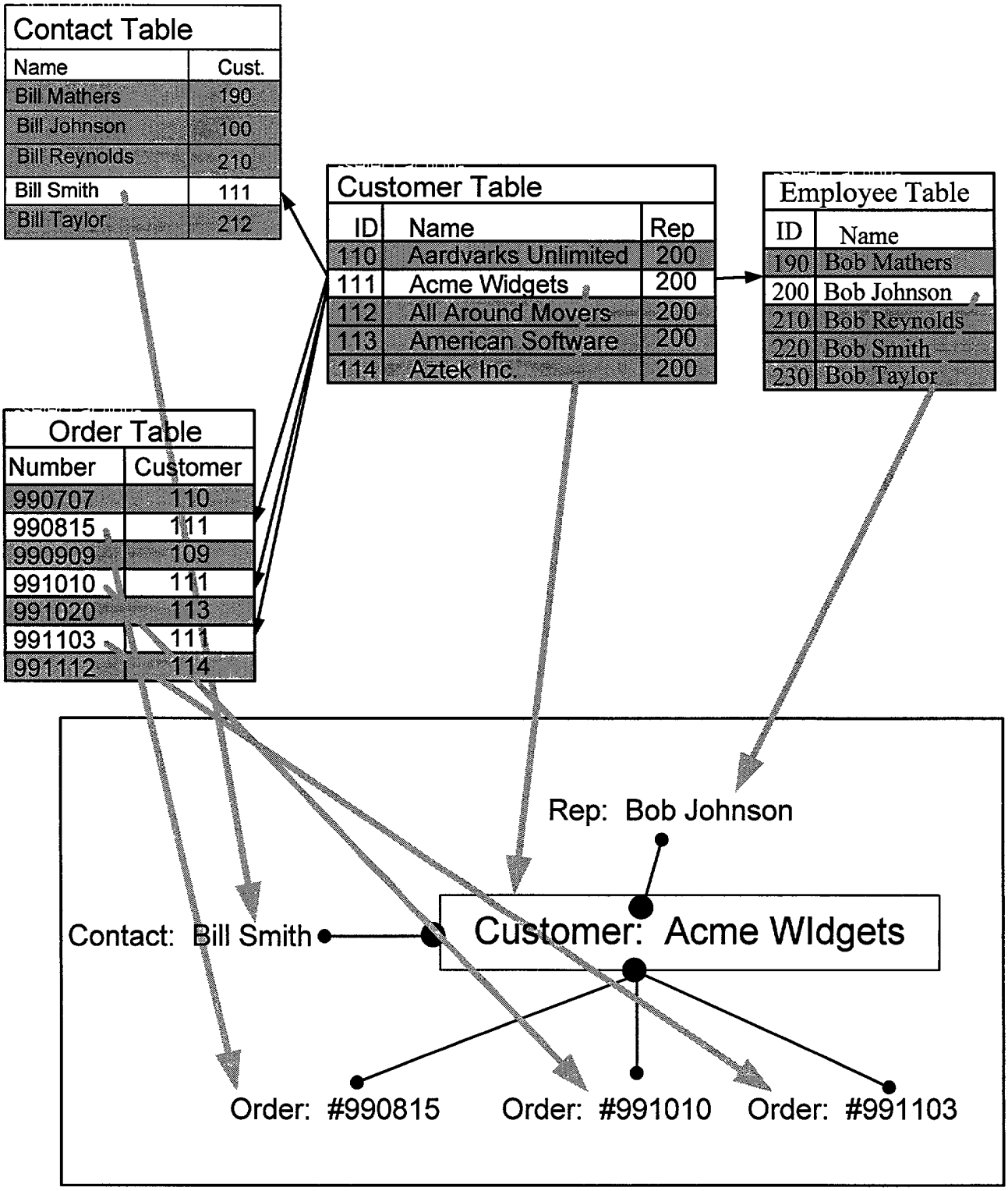
FIG. 35

TOOETT" 2ST2000T



EXAMPLE

FIG. 36



RELATED DATA IN DATABASE TABLES MAPPED TO THEBRAIN

FIG. 37

Express Mail No. EV 759 571 905 US

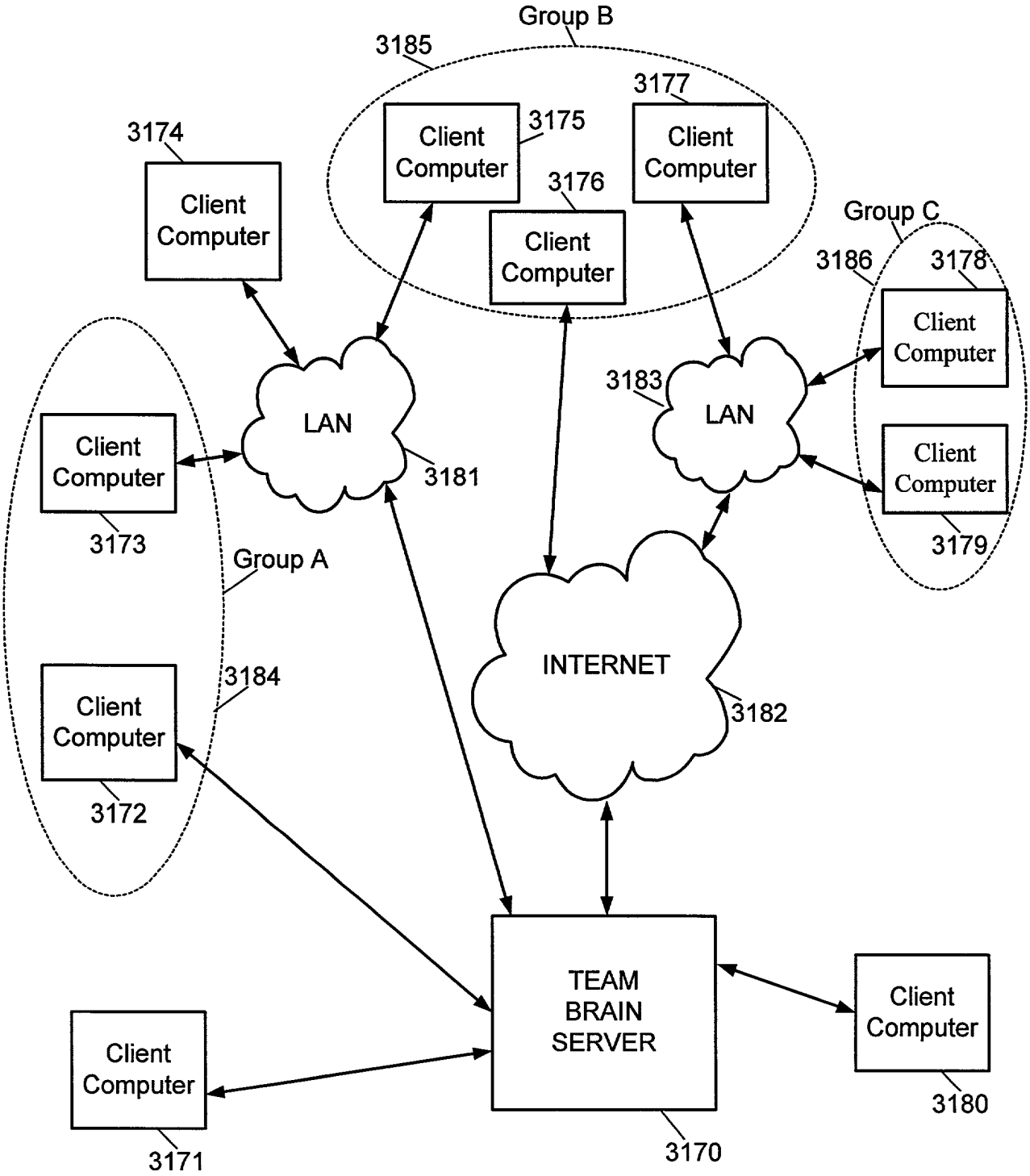


FIG. 38

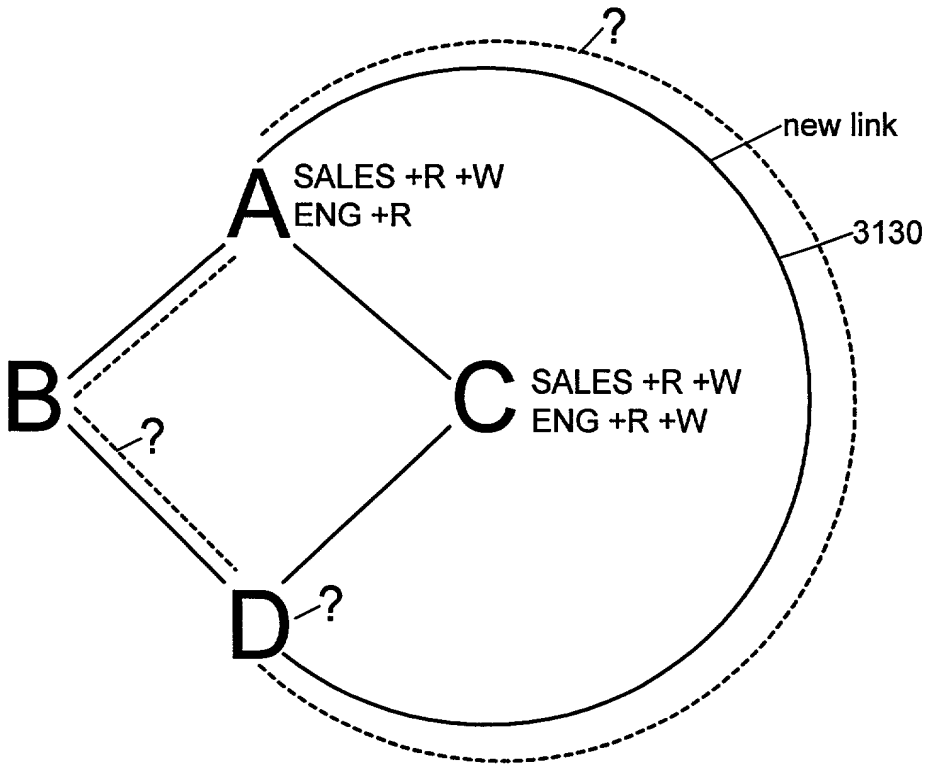
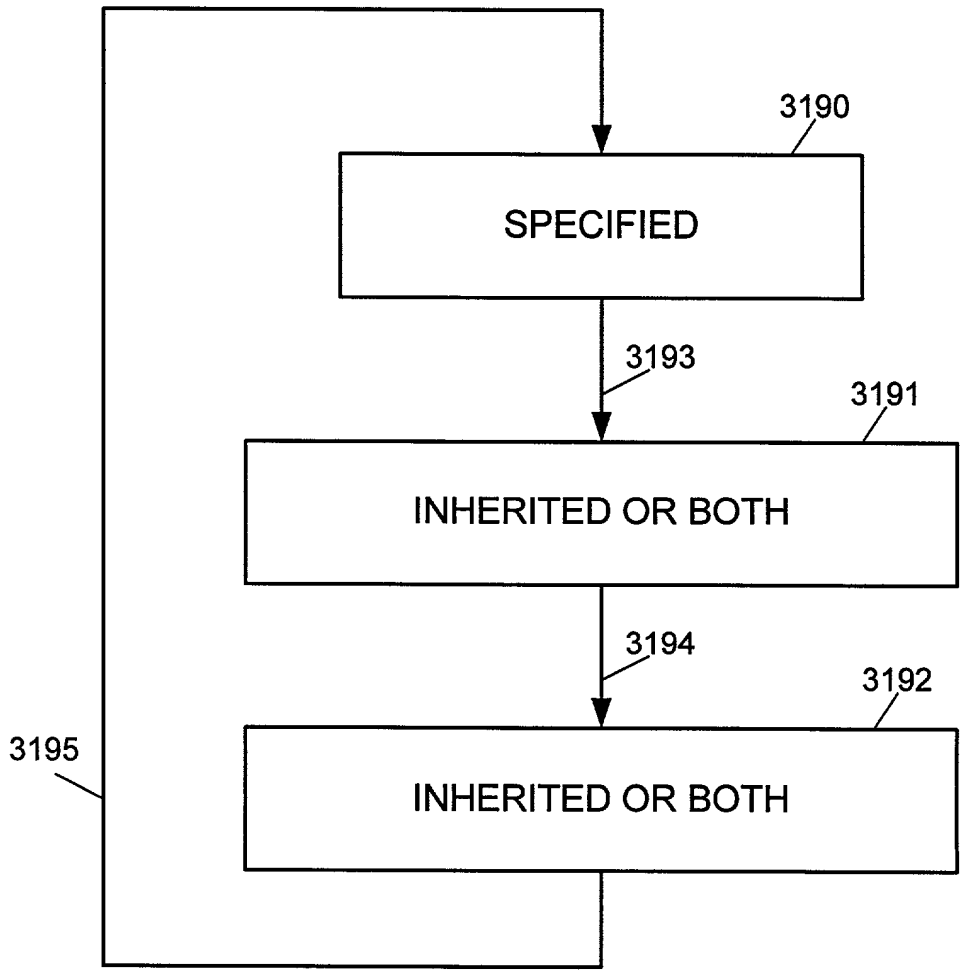
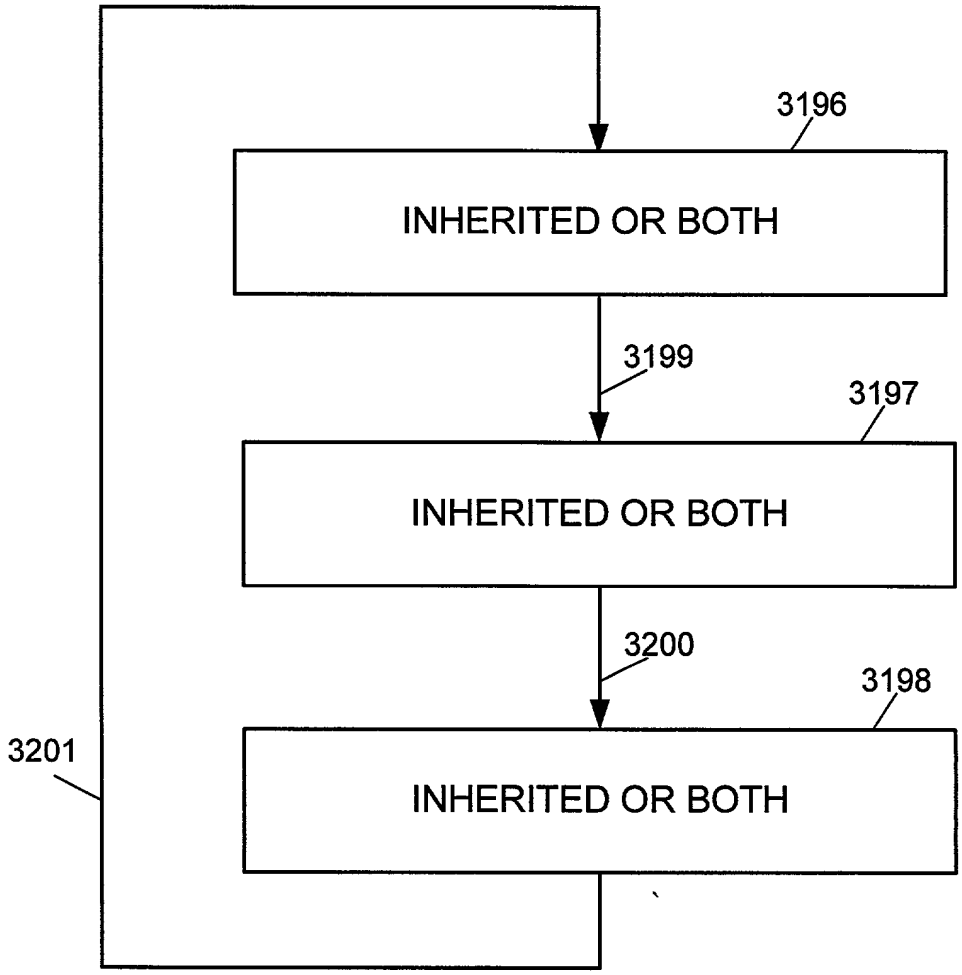


FIG. 39



ALLOWED INHERITANCE

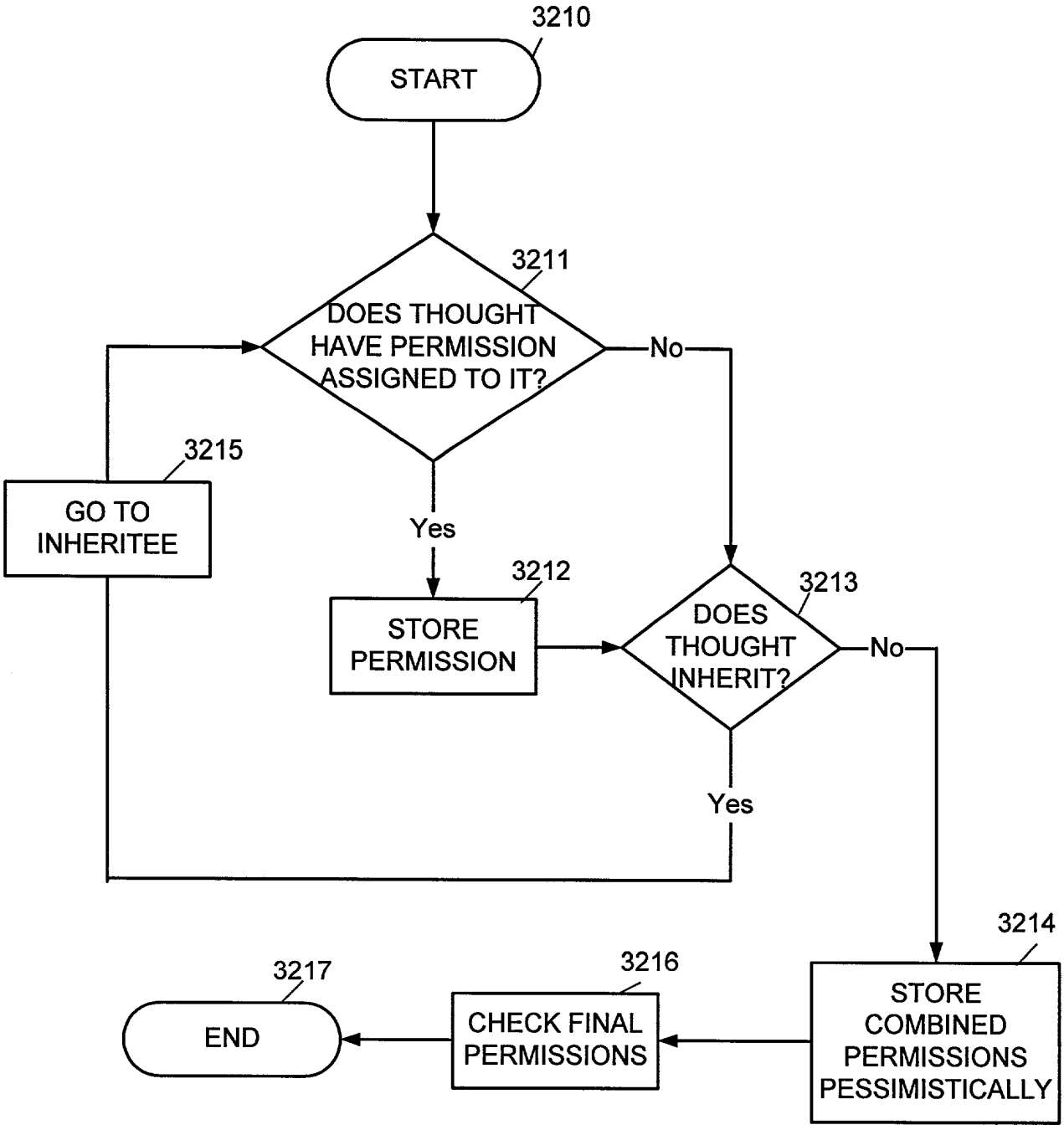
FIG. 40



NOT ALLOWED INHERITANCE

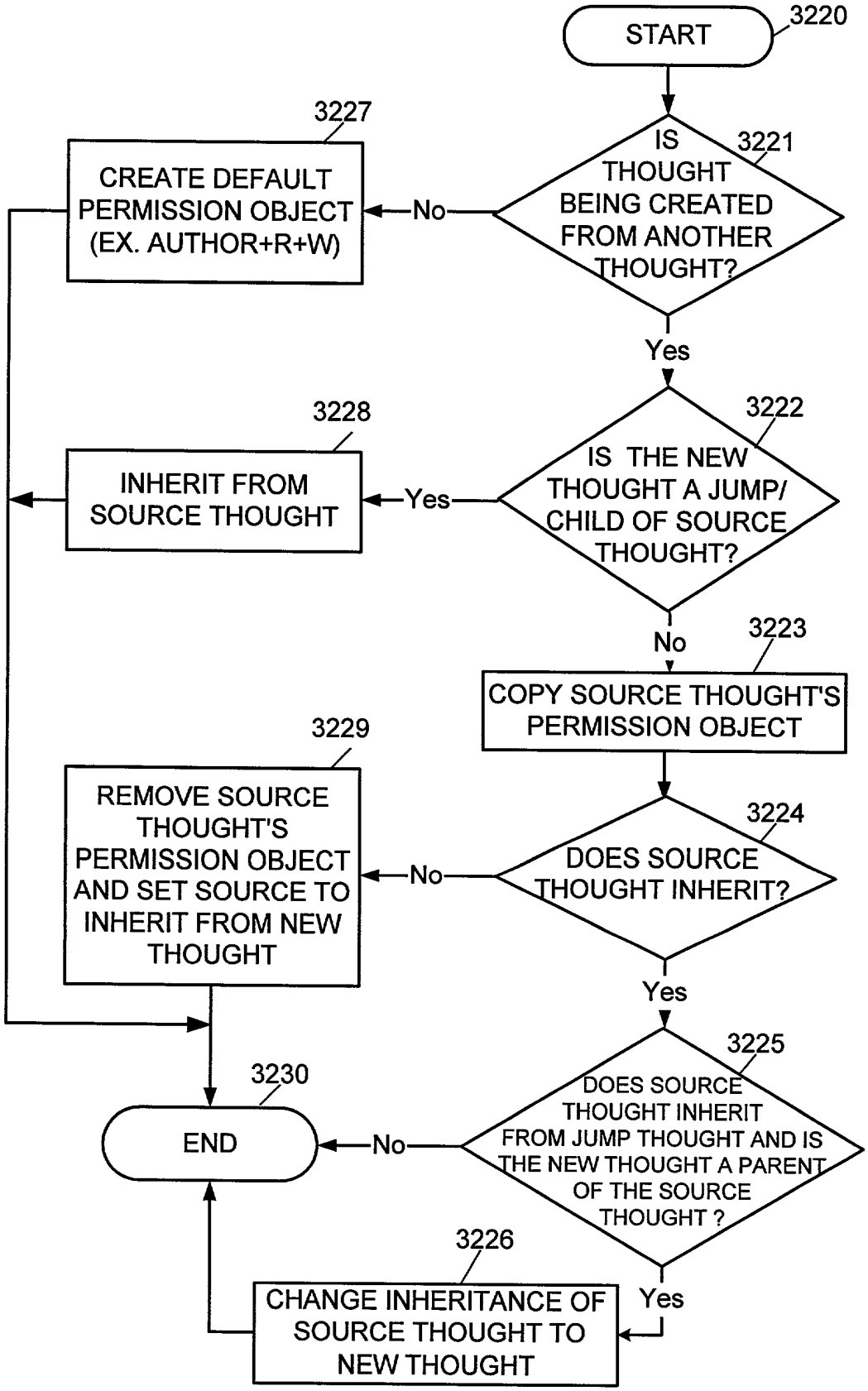
FIG. 41

Patent Application Publication



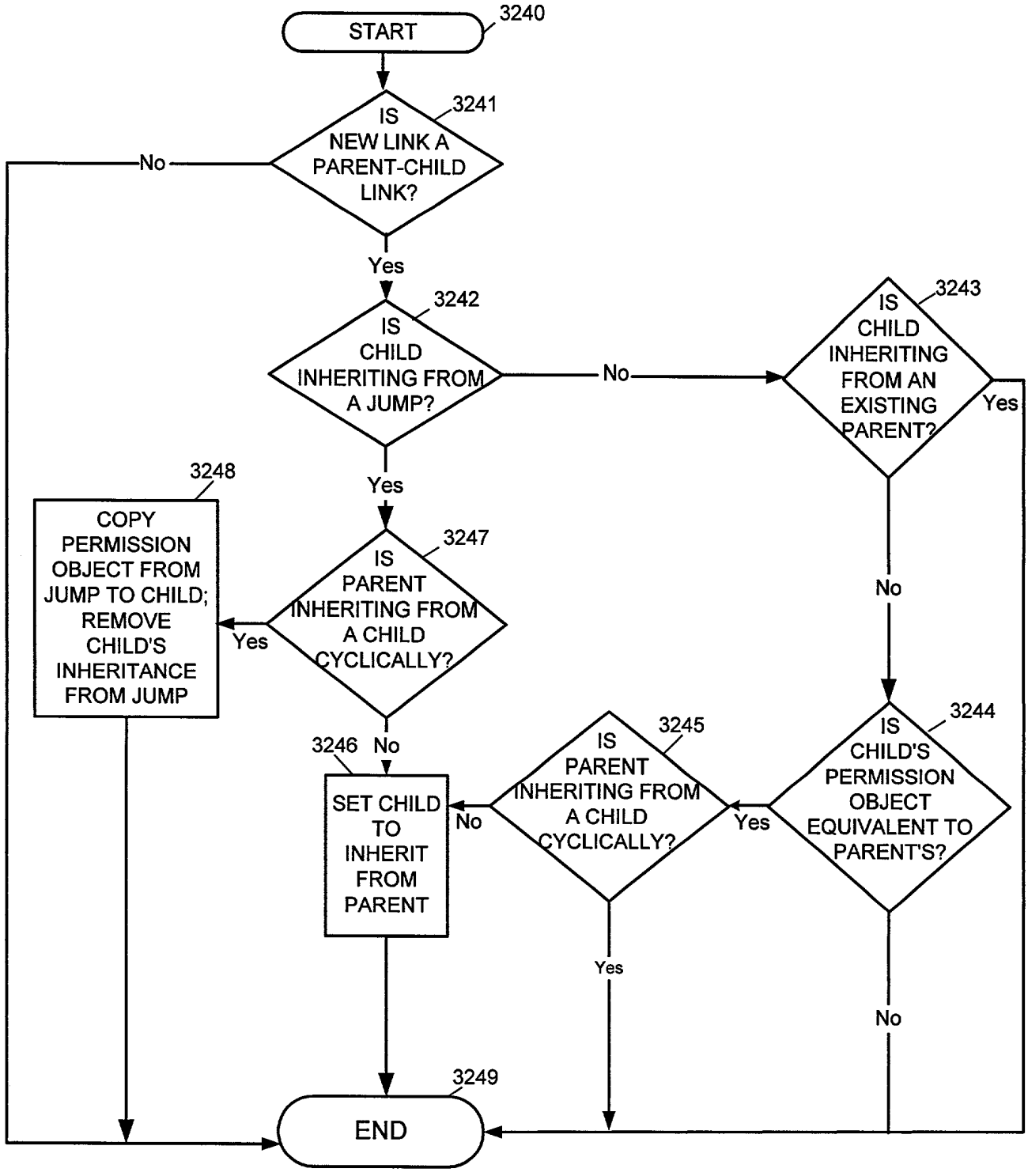
CHECKING PERMISSIONS

FIG. 42



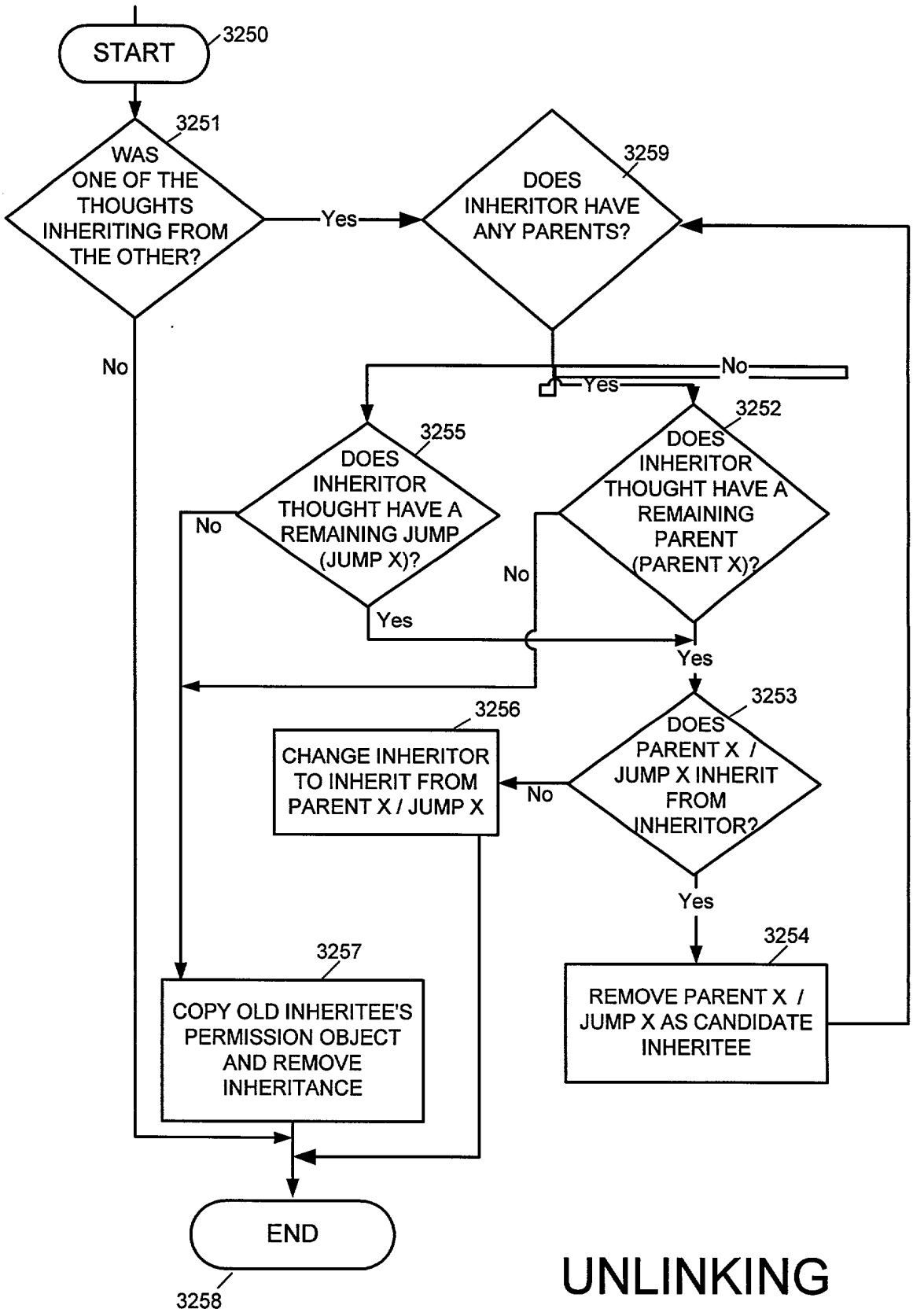
ASSIGNING/INHERITING PERMISSIONS FOR NEW THOUGHT

FIG. 43



CREATING LINKS

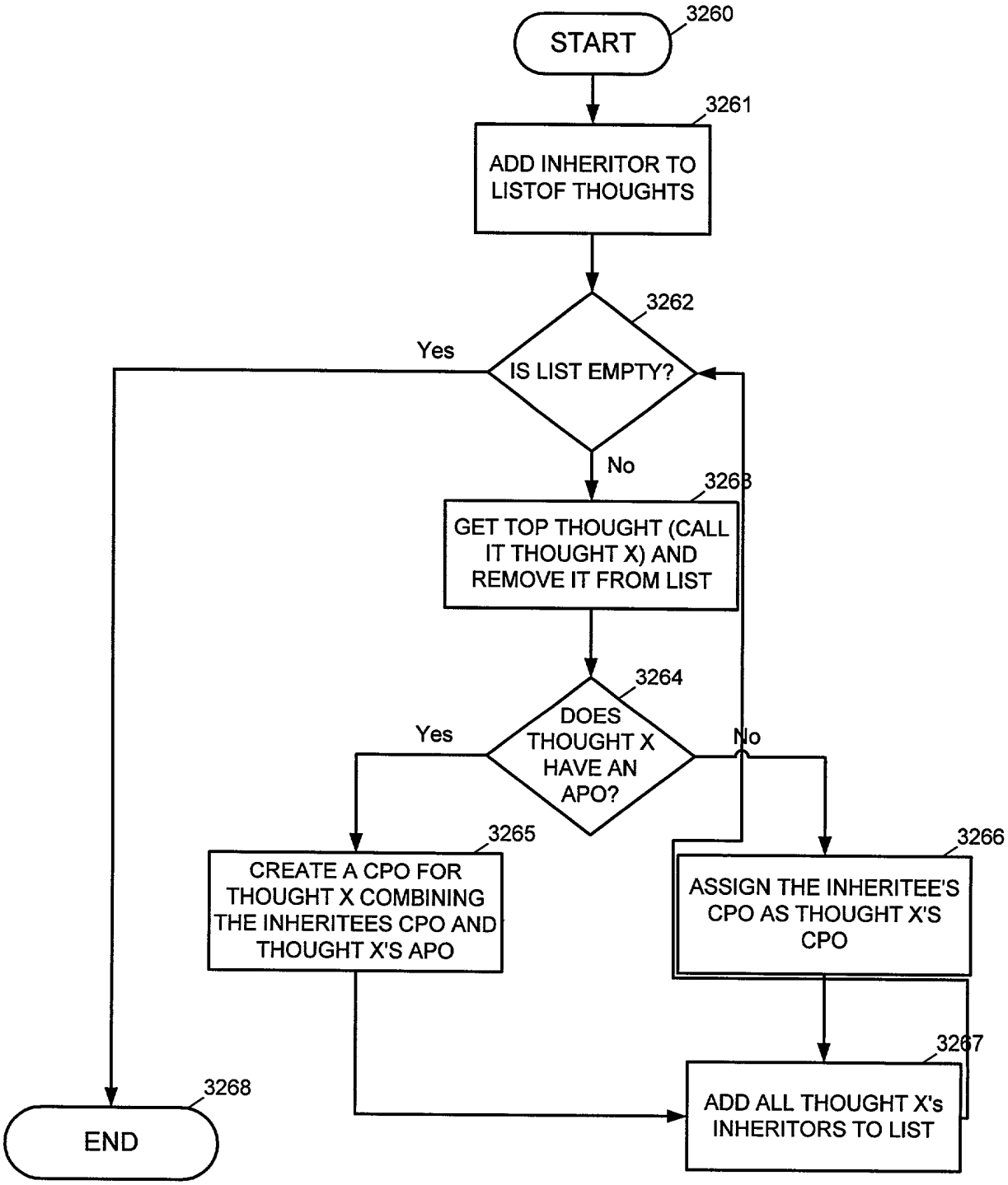
FIG. 44



UNLINKING

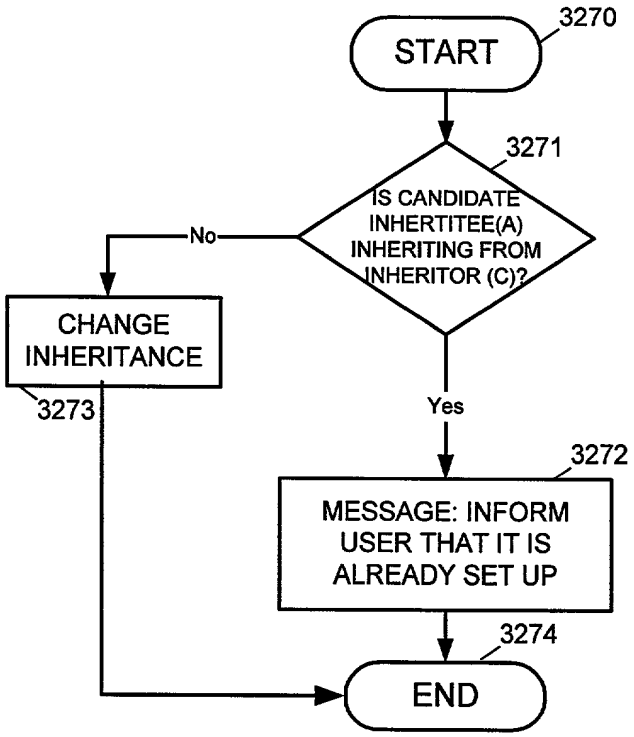
FIG. 45

EXHIBIT 1029, PAGE 1029



PROPAGATION OF PERMISSIONS

FIG. 46



ASSIGNING INHERITANCE

FIG. 47A

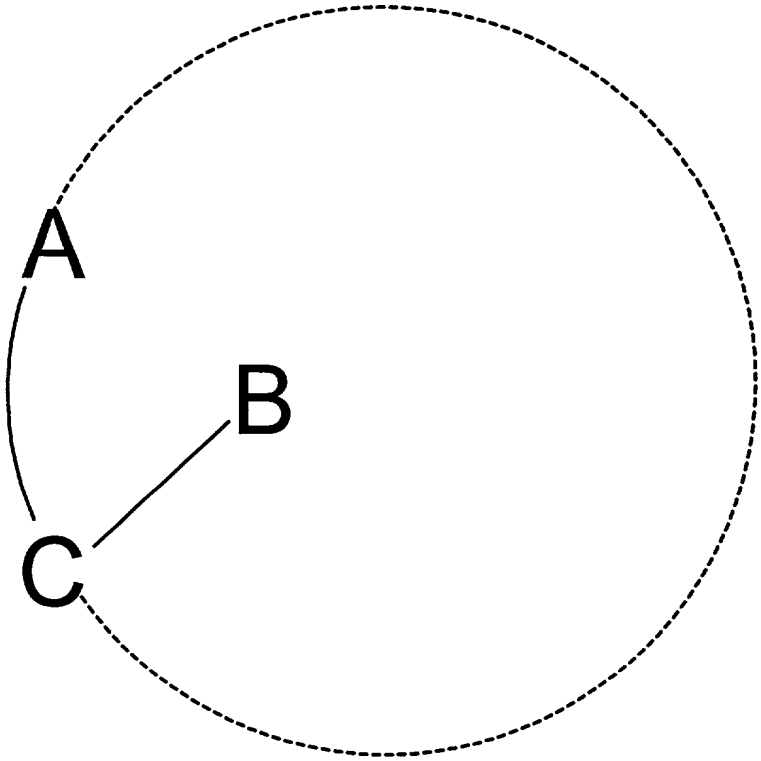
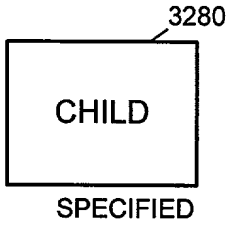
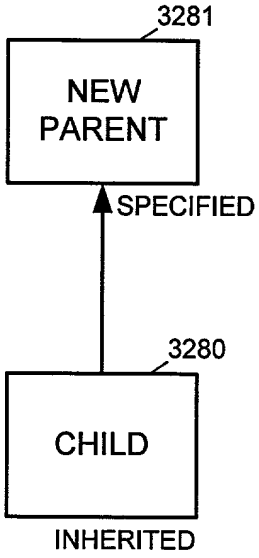


FIG. 47B



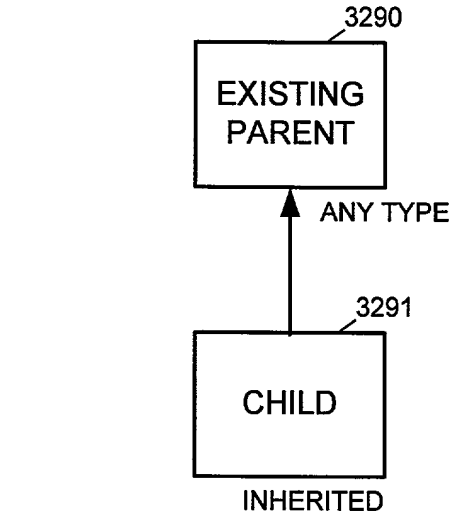
CREATING A PARENT THOUGHT --
NO PARENTS OR JUMPS

FIG. 48A



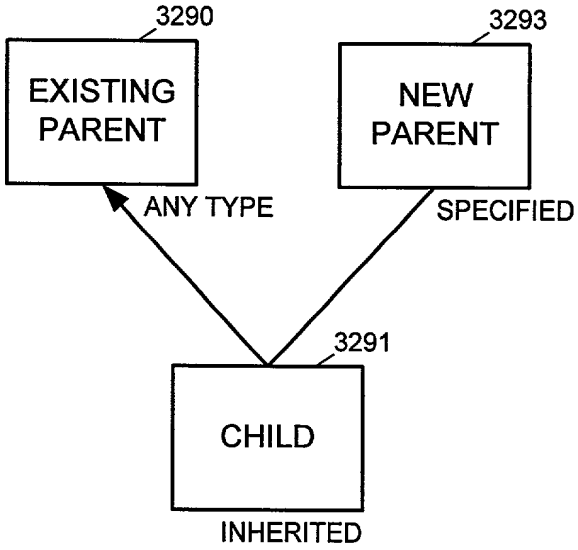
CREATING A PARENT THOUGHT

FIG. 48B



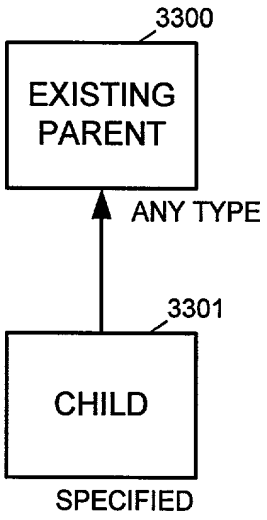
CREATING A PARENT THOUGHT-- ONE OR MORE PARENTS, CHILD INHERITING FROM ONE PARENT

FIG. 49A



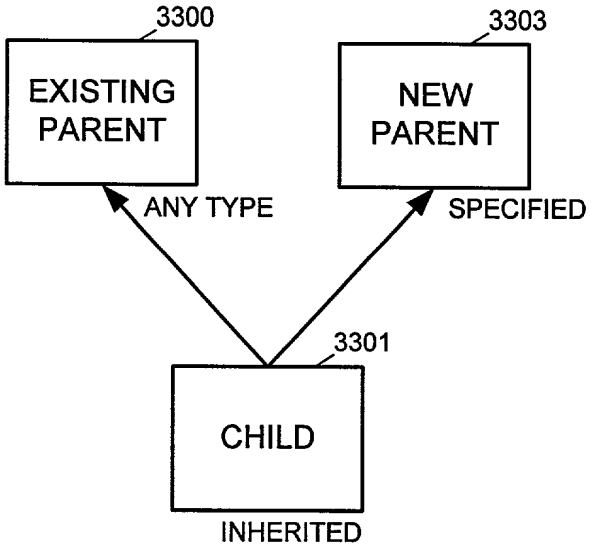
CREATING A PARENT THOUGHT

FIG. 49B



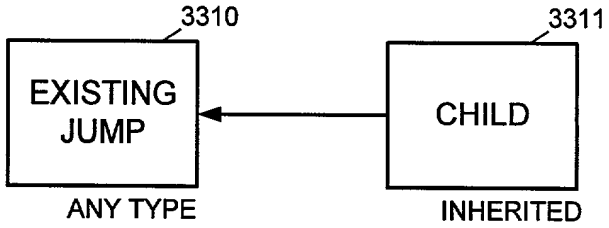
CREATING A PARENT THOUGHT -- ONE OR MORE PARENTS, CHILD PERMISSIONS ARE SPECIFIED

FIG. 50A



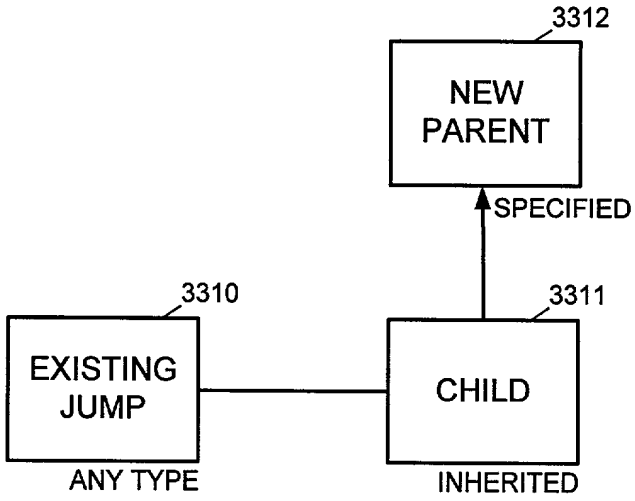
CREATING A PARENT THOUGHT

FIG. 50B



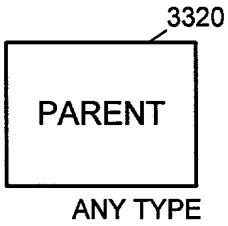
CREATING A PARENT THOUGHT -- NO PARENTS, ONE OR MORE JUMPS

FIG. 51A



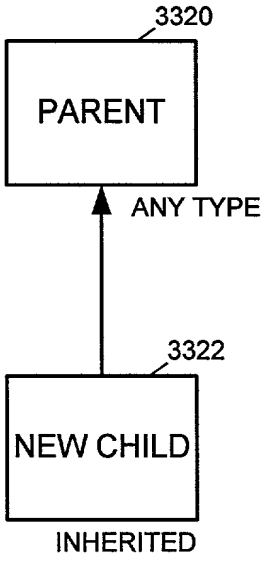
CREATING A PARENT

FIG. 51B



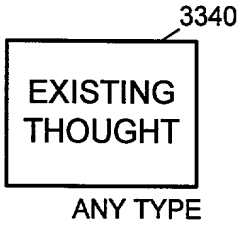
CREATING A CHILD THOUGHT

FIG. 52A



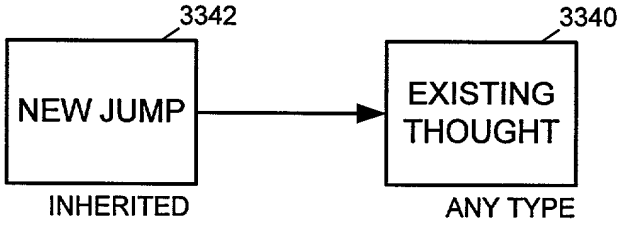
CREATING A CHILD THOUGHT

FIG. 52B



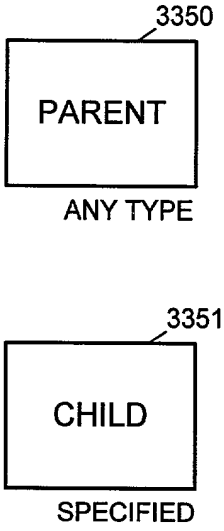
CREATING A JUMP THOUGHT

FIG. 53A



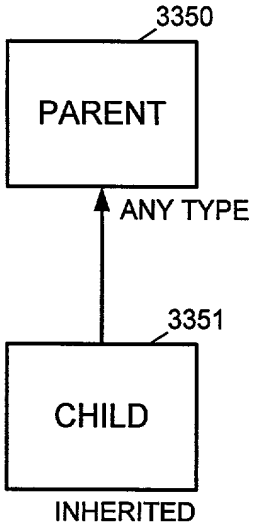
CREATING A JUMP THOUGHT

FIG. 53B



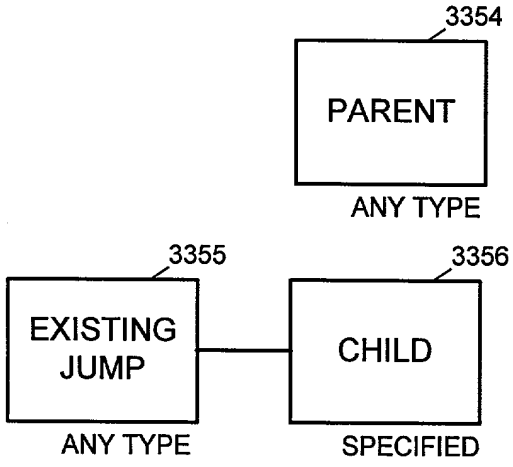
CREATING PARENT-CHILD LINKS

FIG. 54A



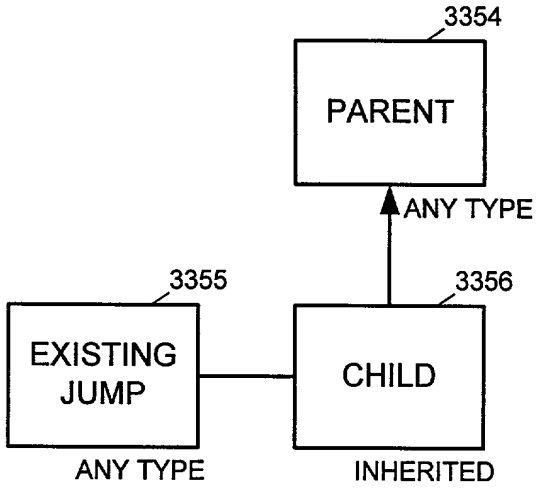
CREATING PARENT-CHILD LINKS

FIG. 54B



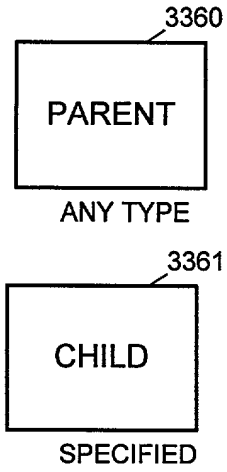
CREATING PARENT-CHILD LINKS

FIG. 54C



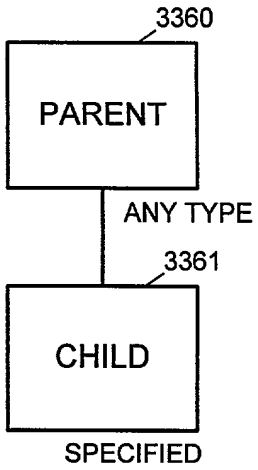
CREATING PARENT-CHILD LINKS

FIG. 54D



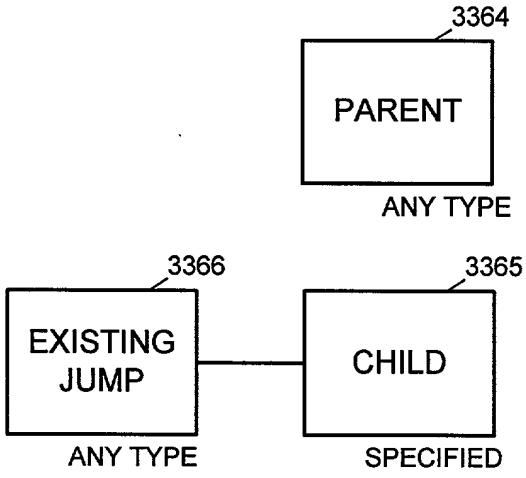
CREATING PARENT-CHILD LINKS

FIG. 55A



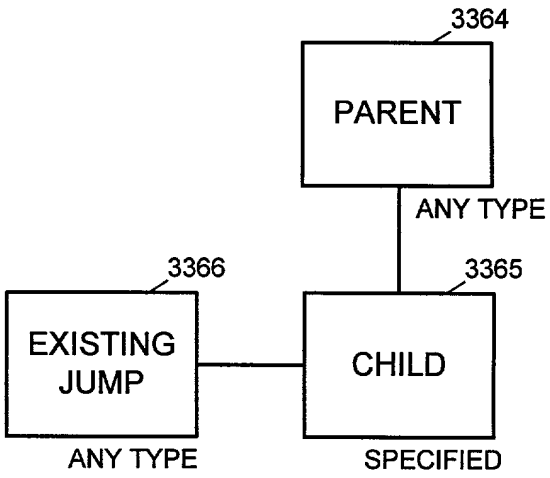
CREATING PARENT-CHILD LINKS

FIG. 55B



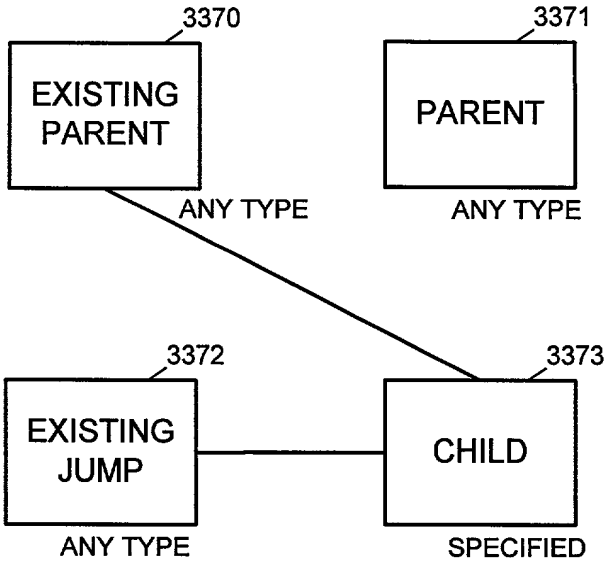
CREATING PARENT-CHILD LINKS

FIG. 55C



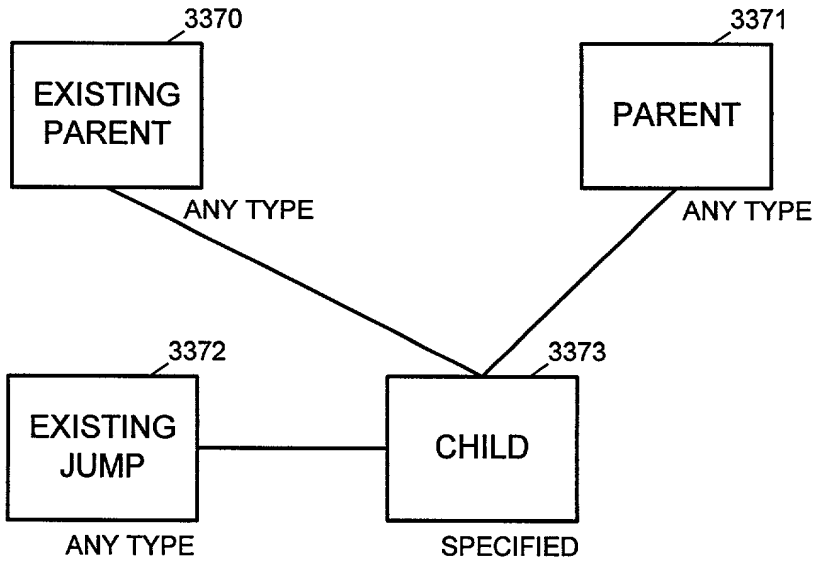
CREATING PARENT-CHILD LINKS

FIG. 55D



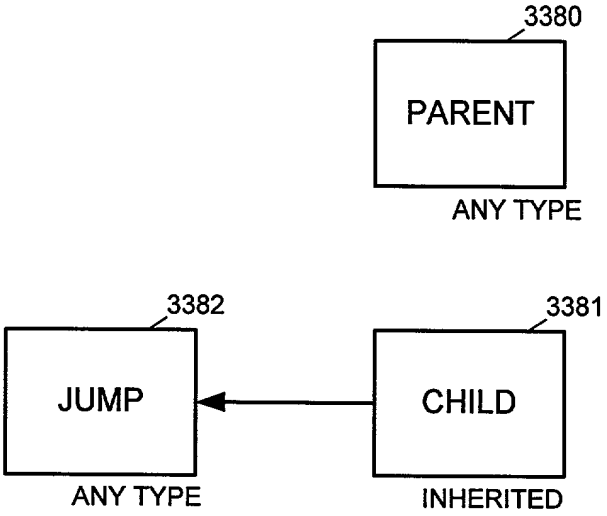
CREATING PARENT-CHILD LINKS

FIG. 55E



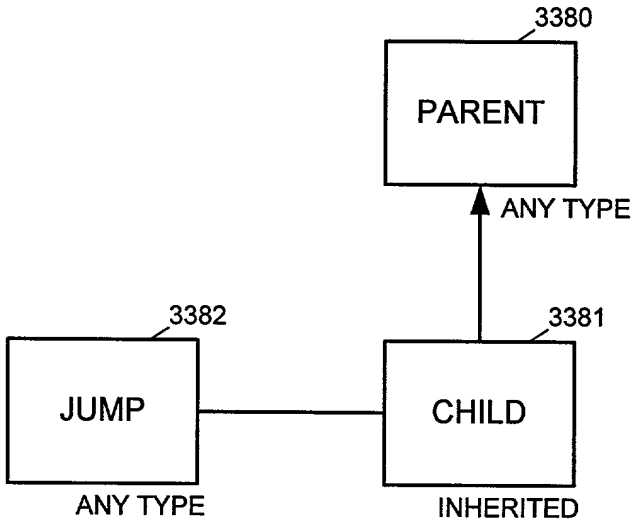
CREATING PARENT-CHILD LINKS

FIG. 55F



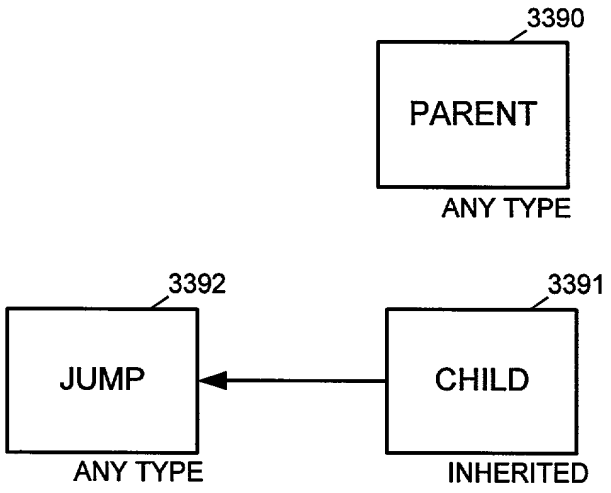
CREATING PARENT-CHILD LINKS

FIG. 56A



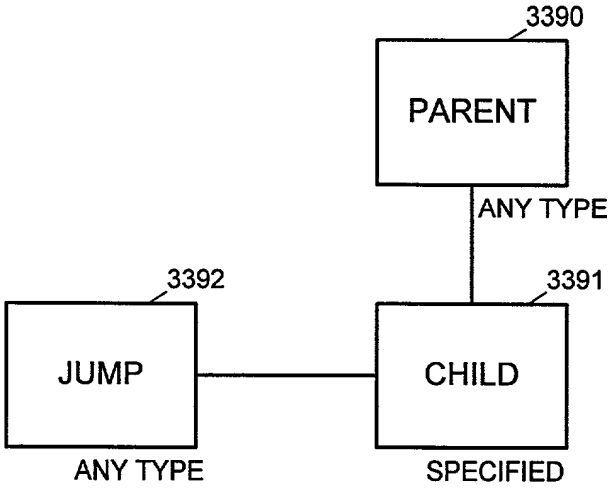
CREATING PARENT-CHILD LINKS

FIG. 56B



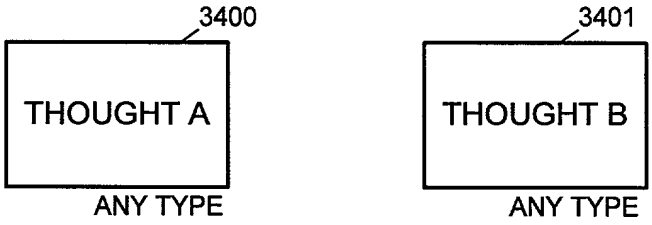
CREATING PARENT-CHILD LINKS

FIG. 57A



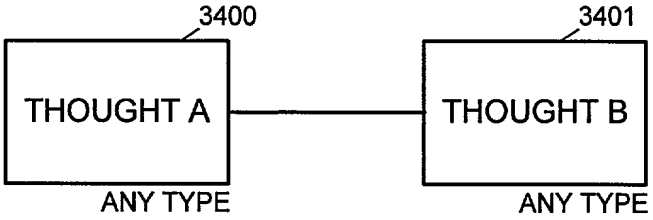
CREATING PARENT-CHILD LINKS

FIG. 57B



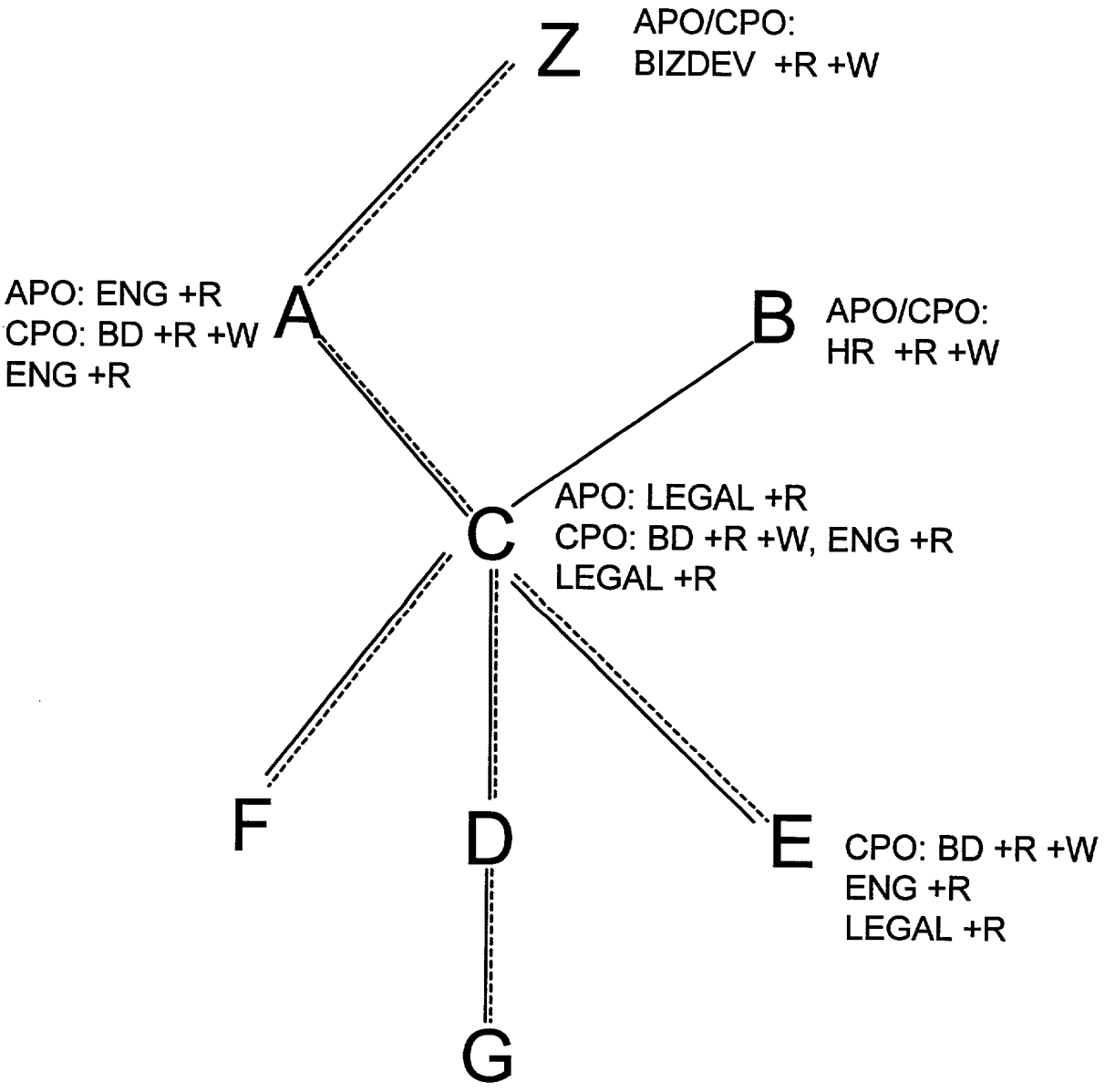
CREATING JUMP LINKS

FIG. 58A



CREATING JUMP LINKS

FIG. 58B



EXAMPLE

FIG. 59

ACTIONS FOR FOLDERS / CABINETS

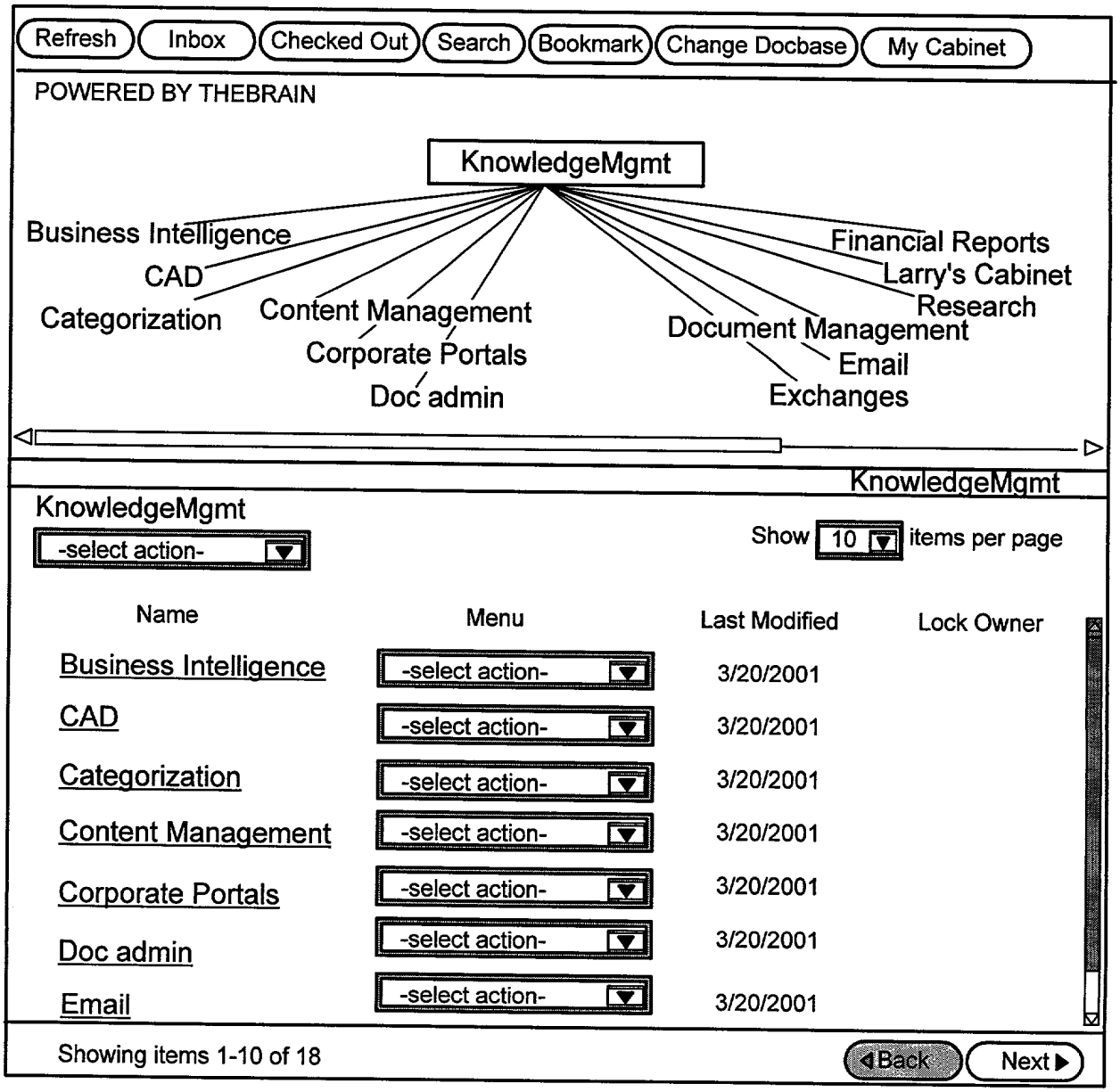


FIG. 60

ACTIONS FOR FOLDERS / CABINETS

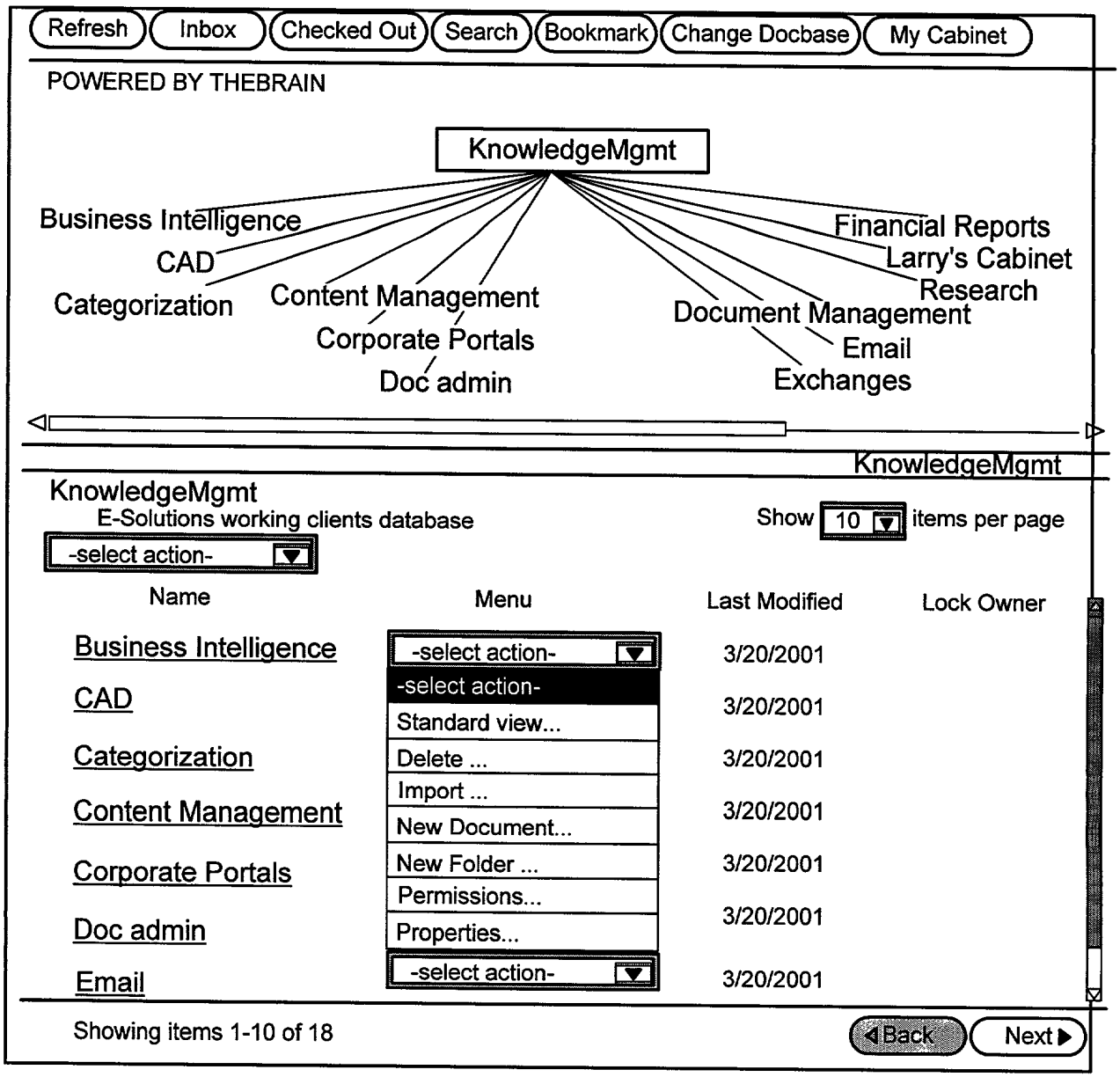


FIG. 61

ACTIONS FOR DOCUMENTS

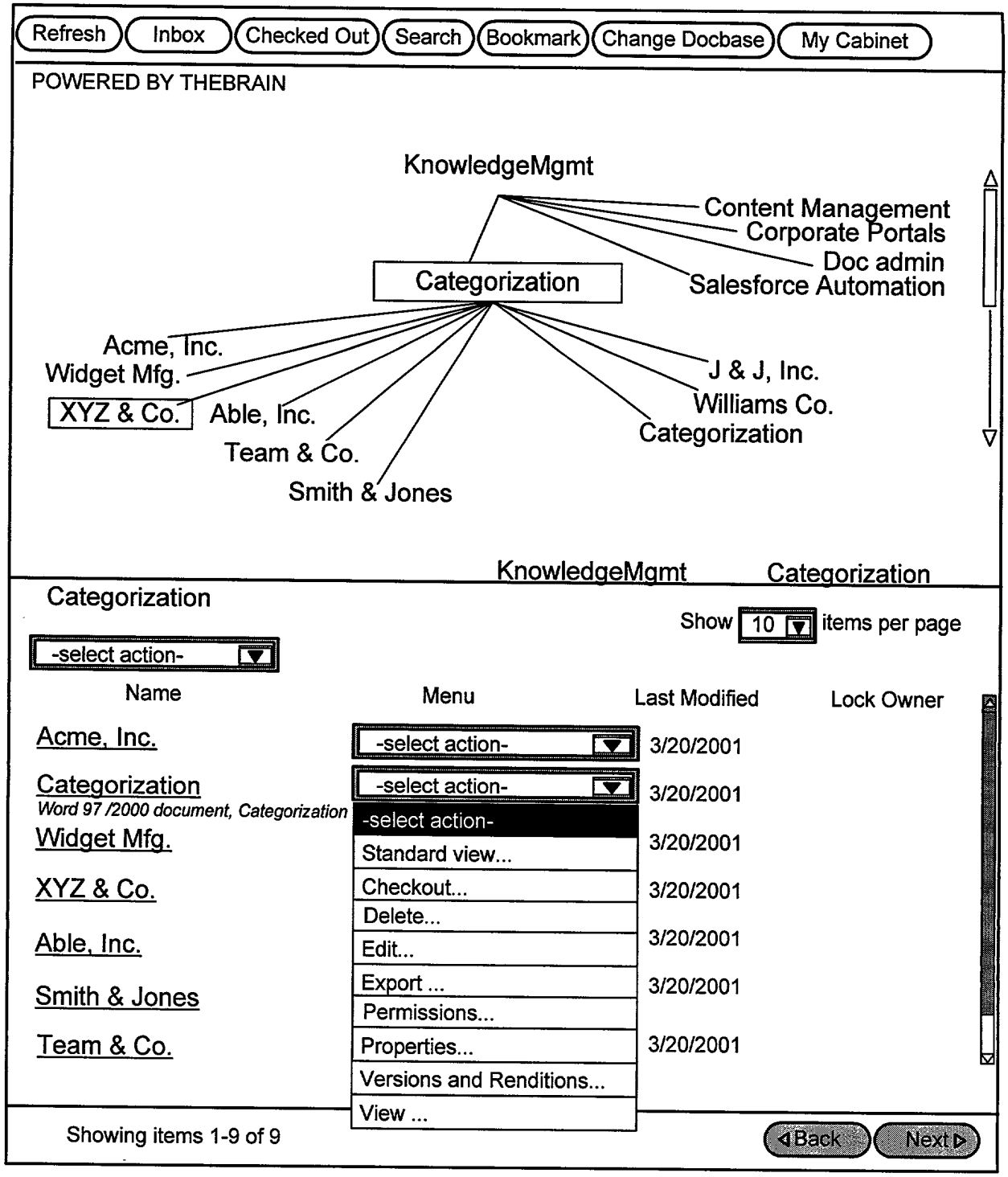


FIG. 62

Figure 63 --- Fully Connected Brain: Modifying an Item, No Conflict

	Client A	Client B	Server
3390	Log in		
3391	Ask server for info about starting thought 1		
3392			Find info about starting thought 1
3393			Return
3394	Display		
3395		Log in	
3396		Ask server for info about starting thought	
3397			Find info about starting thought 1
3398			Return
3399		Display	
3400	Begin modification of data item 1 (thought 1 name, link, property, contents, etc.)		
3401	Send lock request		
3402			Check for existing lock
3403			None found
3404			Lock data item 1 for A
3405	End modification		
3406	Send modification to server and unlock		
3407			Check for other existing lock
3408			None found, confirm and unlock
3409	Display changed data item 1		
3410		Begin modification of data item 1	
3411		Send lock request	
3412			Check for existing lock
3413			None found
3414			Lock item for B
3415		End modification	
3416		Send modification to server and unlock	
3417			Check for existing lock
3418			None found, confirm and unlock
3419		Display changed data item 1	

Patent Application Publication

Figure 64

Fully Connected Brain - Modifying an Item - Conflicts Found

	Client A	Client B	Server
3420	Log in		
3421	Ask server for info about starting thought		
3422			Find info about starting thought
3423			Return
3424	Display		
3425		Log in	
3426		Ask server for info about starting thought	
3427			Find info about starting thought
3428			Return
3429		Display	
3430	Begin modification of data item		
3431	Send lock request		
3432			Check for existing lock
3433			None found
3434			Lock item for A
3435		Begin modification of data item	
3436		Send lock request	
3437			Check for existing lock
3438			Locked by Client A
3439			Inform Client B
3440		Locked - inform user.	
3441	End modification		
3442	Send modification to server and unlock		
3443			Check for other existing lock
3444			None found, confirm
3445	Display changed data		

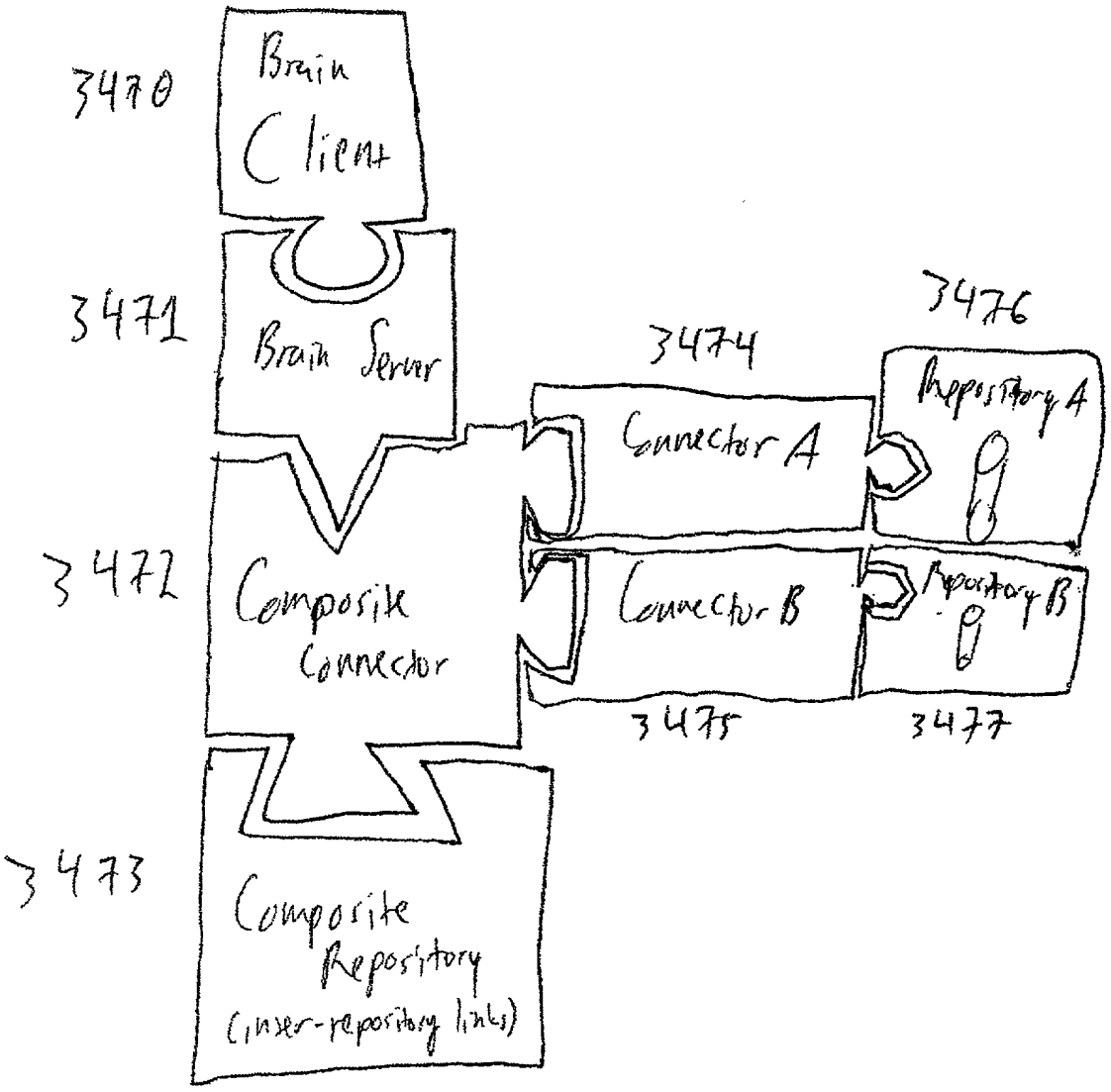
Exhibit 1002, Page 3445

Figure 65

Synchronizing Offline Brains

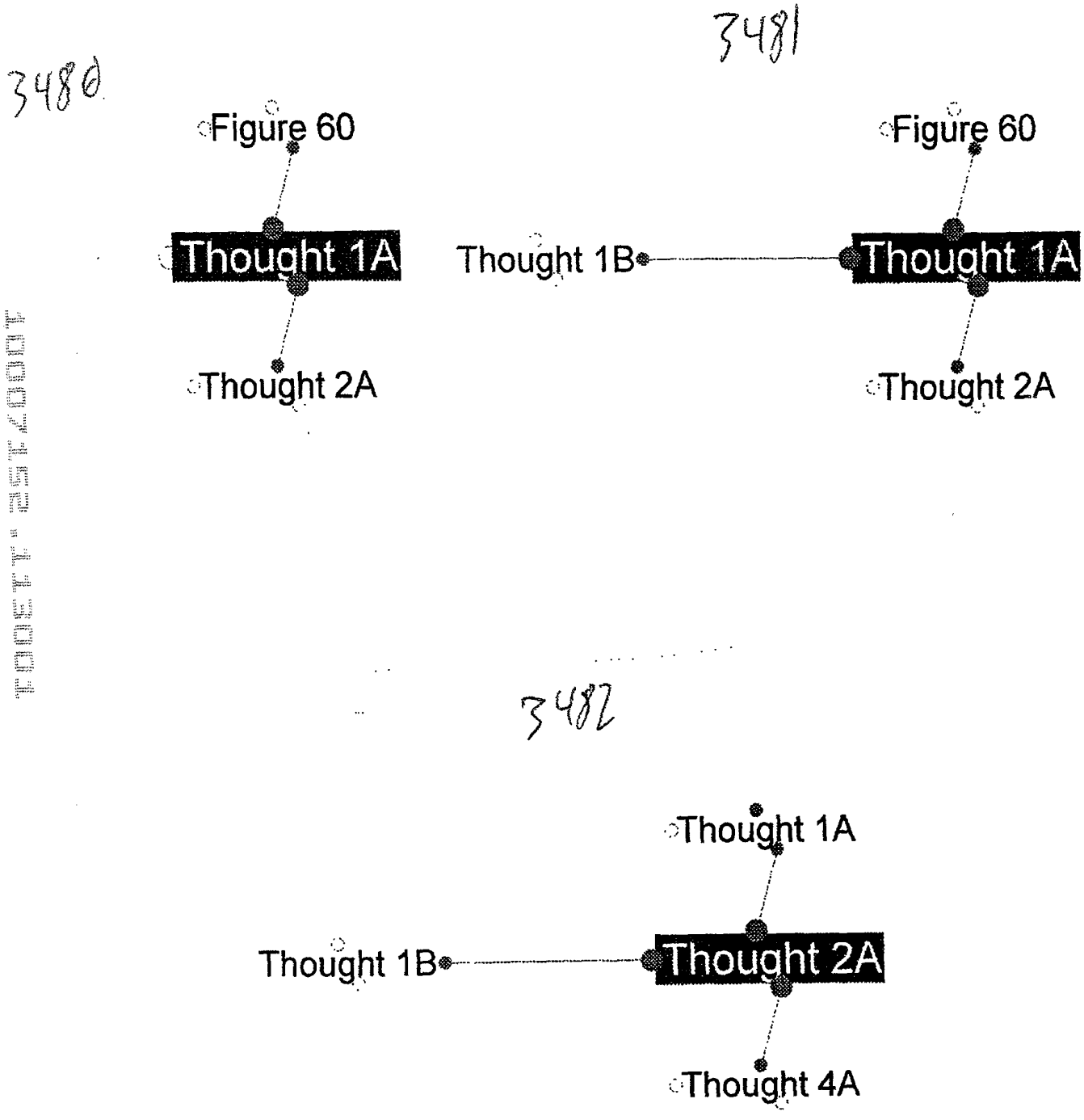
		Client	Server
	3450	**Modifying Data Offline**	
a		Begin modifying data item	
b		Update meta-info (at least data item being modified and time)	
c		Add to list of modified data items	
	3451	**Detecting New or Needed Items**	
a		user selects a thought with links to "missing" thoughts.	
b		add missing thoughts to list of new items.	
	3452	**Updating the Client for Missing Data Items**	
a		Send list of new items	
b			Receive list of new items
c			Add items to list based on synchronization profile
d			Send data for all items on list
e		Receive data and add to local data store	
f		Empty list of new items	
	3453	**Updating Client and Server of new Modifications**	
a		Send list of modified data items with meta-data about modification, but not the data itself	
b		request list of all data items modified since last synchronization	
c			Receive list of modified items
d			Generate list of items modified on the server since last synchonization
e			Detect conflicts between two lists. Decide based on conflict rules
f			When Server Wins Conflict, or When Client Made No Modification: Send data for items
g			When Client Wins, or When Server Made No Modifications: Request data for items.
h		Make modications to items for data received on local data store	
I		Send data for items requested	
j			Receive data items and make modifications on server data store
k		Clear list of modified items	

Figure 66



10097458.1.6004

Figure 67



4006454.1 5E5E-2004

Figure 68

Composite System – Activating a New Thought

	Brain Client (3470)	Brain Server (3471)	Composite Connector (3472)	Connector A (3474)	Connector B (3475)
3490	Thought 1A is active				
3491	Click Thought 2A				
3492		What thoughts are related to Thought 2A?			
3493		Contact Connector			
3494			Translate Thougth ID "2A" into		
3495			Repository A, "ID 2"		
3496				Query Repository A for links to Thought "ID 2"	
3497				Return from Repository: "ID 4, Child"	
3498			Translate link from Connector A "4, Child" as ID "4A, Child"		
3499			Put ID "4A" onto list of related thoughts		
3500			Check Composite Repository for cross-repository links to ID "2A"		
3501			Return from Composite Repository "1B, Jump"		
3502			Add "1B, Jump" to list of related thoughts		
3503			Return List		
3504		Return List			
3505	Activate 2A as new central thought, with Thought ID "1B" as a Jump, and Thought ID "4A" as a Child.				
3506		Repeat Procedure for Siblings, Grandchildren, if applicable.			

Figure 69

Composite System -- Making a Link

	Brain Client (3470)	Brain Server (3471)	Composite Connector (3472)	Connector A (3474)	Connector B (3475)
3510	Thought 1A is Actve				
3511	User draws jump from Thought 1A to Thought 1B.				
3512		Make Jump Link Thought 1A to Thought 1B			
3513		Contact Composite Connector			
3514			Translate Thougth ID "1A" into Repository A, "ID 1"		
3515			Translate Thougth ID "1B" into Repository B, "ID 1"		
3516			Same Repository?		
3517			No ==>Add link to Composite Repository.		
3518			Yes==> Contact Connector A or Connector B, as fit.		
3519		Success Reported			
3520	Modify Display to Plex 3481				

Exhibit 2002 - Page 0135

Figure 70
Composite System -- Creating a Thought

	Brain Client (3470)	Brain Server (3471)	Composite Connector (3472)	Connector A (3474)	Connector B (3475)
3521	Thought 1A is Active				
3522	User creates child Thought				
3523		Create Child			
3524		Contact Composite Connector			
3525			Translate Thougth ID 1A into Repository A "ID 1".		
3526			Any special rules for children of thoughts of the type of "ID 1"?		
3527			No ==> Send to Connector A. Yes ==> Send to Repository B.		
				If requested, add to Repository A. Return "success" and thought type to Client.	
3528					If reuqested, add to Repository B. Update Composite Connector
3529			Return results to Server; Update Composite Repository if Child was placed in different repository than parent.		
		Return Results			
	Modify Display to Plex 3484, conform contents to any special rules.				

Figure 71

Brain to Brain Links System

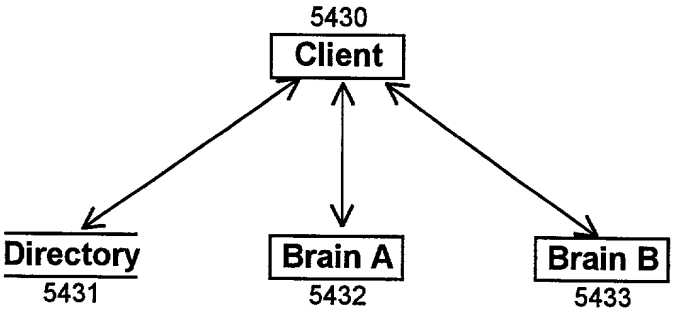


Figure 72

Activating a Thought

	Client 5430	Directory Server 5431	Brain A Server 5432	Brain B Server 5433
5440	Plex 3482. Click 2A			
5441	Check local directory cache for location of Brain Server A			
5442	Found			
5443	Request to Brain A re: thought info for thought 2			
5444			Query DB for info and links re thougth 2.	
5445			Links from 2 to 1B is found	
5446			Return list	
5447	Check results for for inter-Brain links.			
5448	Find 1B			
5449	Check local directory cache for location of Brain Server B.			
5450	Not found.			
5451	Query: Where is B?			
5452		Look up location of B		
5453		Return		
5454	Add B's location to local directory cache			
5455	Request to Brain B for thought info for thought 1			
5456				Query for info
5457				Return
5458	Change to display Plex 3482			

Exhibit 1029, page 1828

Figure 65

Figure 73

Linking a Thought (assumes both are onscreen)

	Client 5430	Directory Server 5431	Brain A Server 5432	Brain B Server 5433
5460	Link 1A and 1B (jump) One thought may be active in the plex the other already as a pin or in the past thought list (both are multi-brain enabled)			
5461	Check local dir cache for location of A			
5462	Found			
5463	Send message to A describing link from 1 to 1B (relationship, thought id, brain id "jump, unique brain id of B")			
5464			Store link from 1 to 1B	
5465	Check dir cache for B			
5466	Found			
5467	Send message to B describing link from 1 to 1A ("jump, unique brain id of A")			
5468				Store link from 1 to 1B
5469	Display link			

Figure 74

Creating a Thought

	Client 5430	Directory Server 5431	Brain A Server 5432	Brain B Server 5433
5470	Create new thought from 1A (jump)			
5471	Check local dir cache for location of A			
5472	Found			
5473	Send message to A describing new thought from 1 (relationship, name/content "jump, alan")			
5474			Create new thought and store content	
5475			Store link between id 1 and new id	
5476			Return new thought and link info	
5477	Display			

Express Mail No. EV339141925US

Figure 75

Finding Other Brains

	Client	Directory Server	Brain C Server
5480	Request list of Brains or search for a Brain		
5481		Look for Brains that match request	
5482		Return results (Brain name, id, location)	
5483	Add locations to cache		
5484	Display results in a window.		
5485	User selects a Brain (Brain C)		
5486	Look up Brain in cache		
5487	Ask for starting thought		
5488			Query DB for info and links.
5489			Return
5490	Display		

Finding Thoughts in Other Brains

	Client	Directory Server	Brain A Server	Brain B Server	Brain C Server
5491	Search for a Thought (may specify a list of Brains to search in via a list obtained from the directory server, or search a default set)				
5492	Ask directory server to perform search				
5493		Query each Brain server			
5494			Check for matching thoughts	Check for matching thoughts	Check for matching thoughts
5495			Return list	Return list	Return list
5496		Compile results into a single list with all info (Brain id & location, thought info & id)			
5497	Add locations to cache				
5498	Display results in a window.				
5499	User selects a Thought				
5500	Activate thought				

Figure 76

File Structure for Multiple Files Per Thought

Single Document Architecture

5510 *The Headcase*

	Thought ID	*Name*	*Type*	*Property*	*Document*
5511	1	Thought 1	-	-	doc_r
5512	2	Thought 2	-	-	doc_n
5513	3	Thought 3	-	-	doc_m
5514	4	Thought 4	-	-	doc_q

Multiple Document Architecture

5515 *Thought Table*

	Thought ID	*Name*	*type*	*property*
5516	1	Thought 1	-	-
5517	2	Thougth 2		-
5518	3	Thought 3	-	-
5519	4	Thought 4	-	-

5520 *Link Table*

	Thought ID	*Doc ID*
5521	1	r
5522	1	n
5523	2	m
5524	3	q
5525	4	z
5526	4	n

5527 *Document Table*

	Doc ID	*Document Location*
5528	r	c:\doc_r
5529	n	c:\doc_n
5530	m	c:\doc_m
5531	q	c:\doc_q
5532	z	c:\doc_z

Figure 77

Before User Interaction

After User Interaction

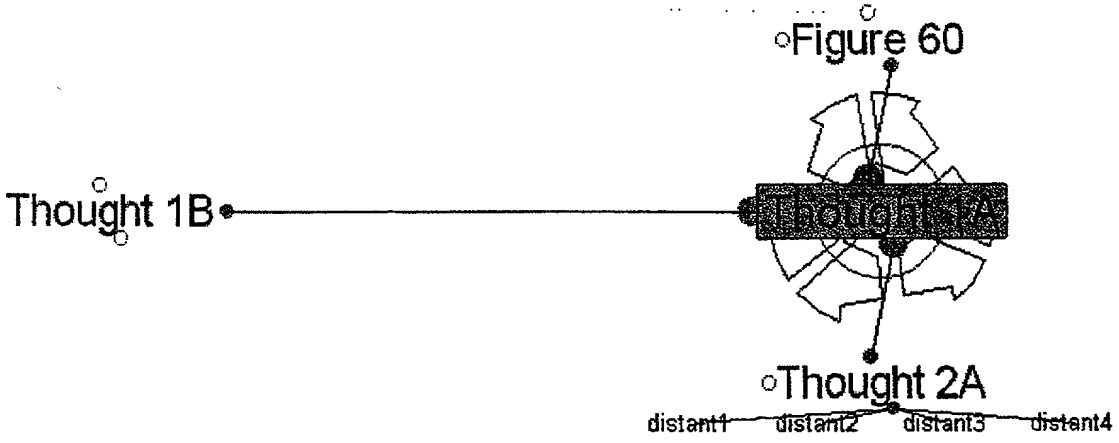


Exhibit 1 Page 55 of 94